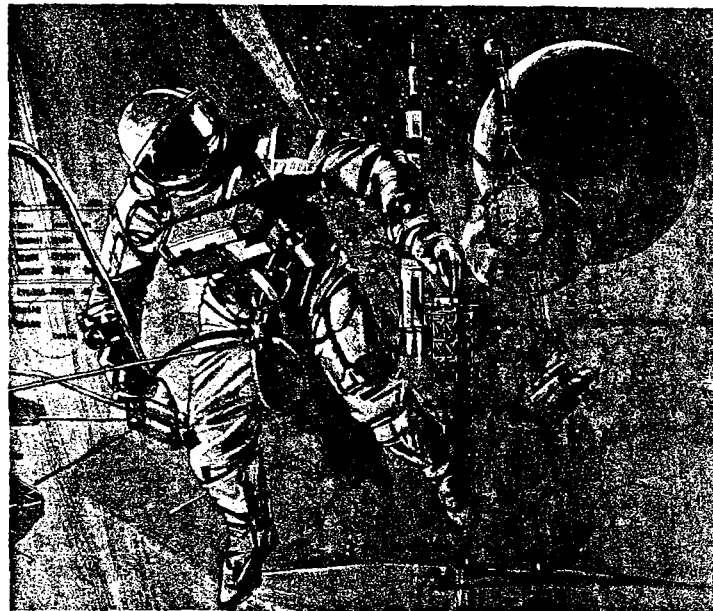


Personal Computer
MZ-800

TECHNICAL REFERENCE
MANUAL



SHARP®

Personal Computer
mz-800

**TECHNICAL REFERENCE
MANUAL**

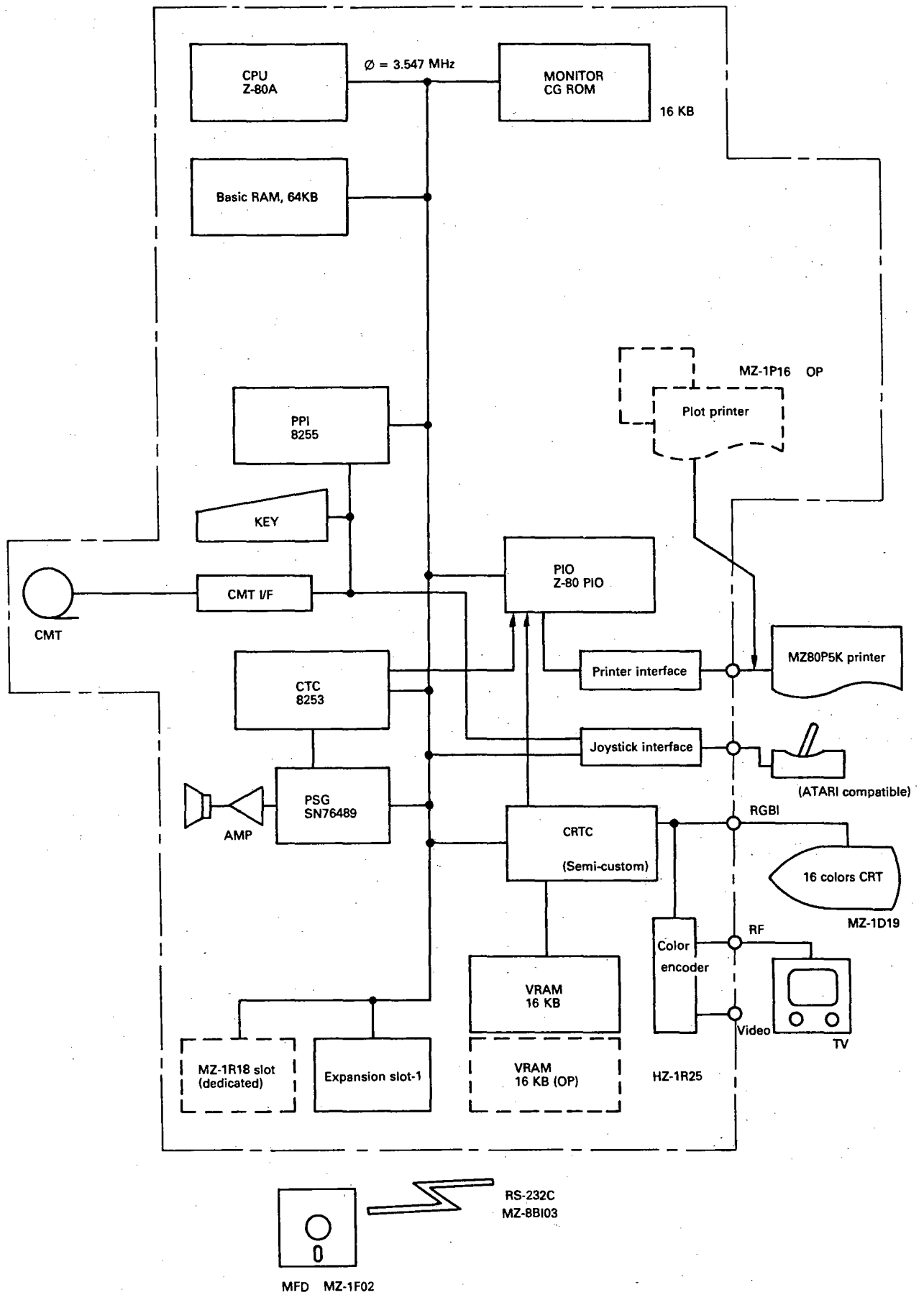
MZ-800 HARDWARE

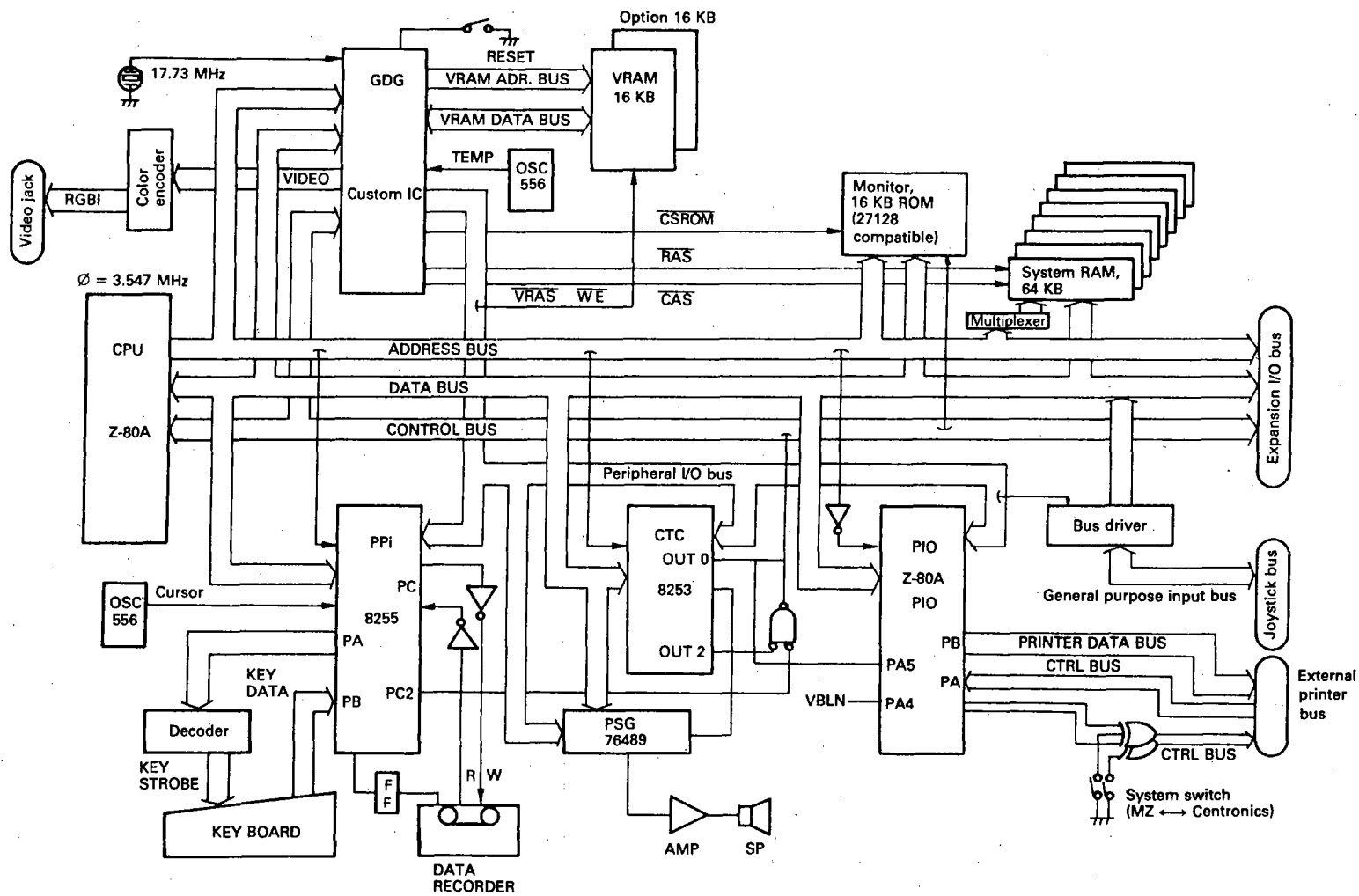
1	System description	1
2	Block diagram	2
3	Memory map	3
4	Custom LSI	6
4-1	Memory controller	6
4-2	I/O controller	6
4-3	Clock generator and timing generator	9
4-4	Display address generator	9
4-5	Scroll	10
4-6	VRAM data input/output circuit	12
4-7	Register functions	17
4-8	Pallet	22
4-9	CRTC register map	23
4-10	ROM configuration	24
5	8255 Programmable Peripheral Interface	24
6	8253 Programmable Interval Timer	28
7	Printer interface	28
8	Programmable sound generator	31
9	Joystick	32
10	Power supply	33
10-1	Block diagram	33
10-2	Operational description	34
10-3	Maintenance	34
10-4	Problem determination and sequence	34
11	MZ-1P16	35
11-1	Installation	35
11-2	Block diagram	36
11-3	At power on	36
11-4	Colour change operation	36
11-5	Pen change operation	36
11-6	Pen exchange method	37
11-7	Stepping motor driving signal	37
11-8	Colour position detector	37
11-9	Character set	37
11-10	Colour plotter printer control LSI	38
11-11	Interfacing with the MZ-800	39
11-12	Block diagram	39
11-13	Colour plotter printer control codes	40
11-14	Graphic mode commands	41
11-15	Specification	42
11-16	Circuit	42
12	CPU P.W.B. Circuit	43

MZ-800 SOFTWARE

1	Character generator (C.G.)	45
1-1	How to modify C.G. ROM	45
1-2	C.G. Table	46
1-3	C.G. ROM dump list	48
2	MZ-800 Monitor subroutines	52
3	Assemble list	67
3-1	Monitor <1Z-013B>	67
3-2	MZ Disk control <9Z-504M-V1.0C>	100
3-3	Mini-floppy disk control	111
3-4	ROM Monitor	117
4	Z-80 Programming reference	138
4-1	Z-80 Status indicators (Flags)	138
4-2	Notation	140
4-3	Z-80 Instruction set	141
	INDEX	173

1 System description

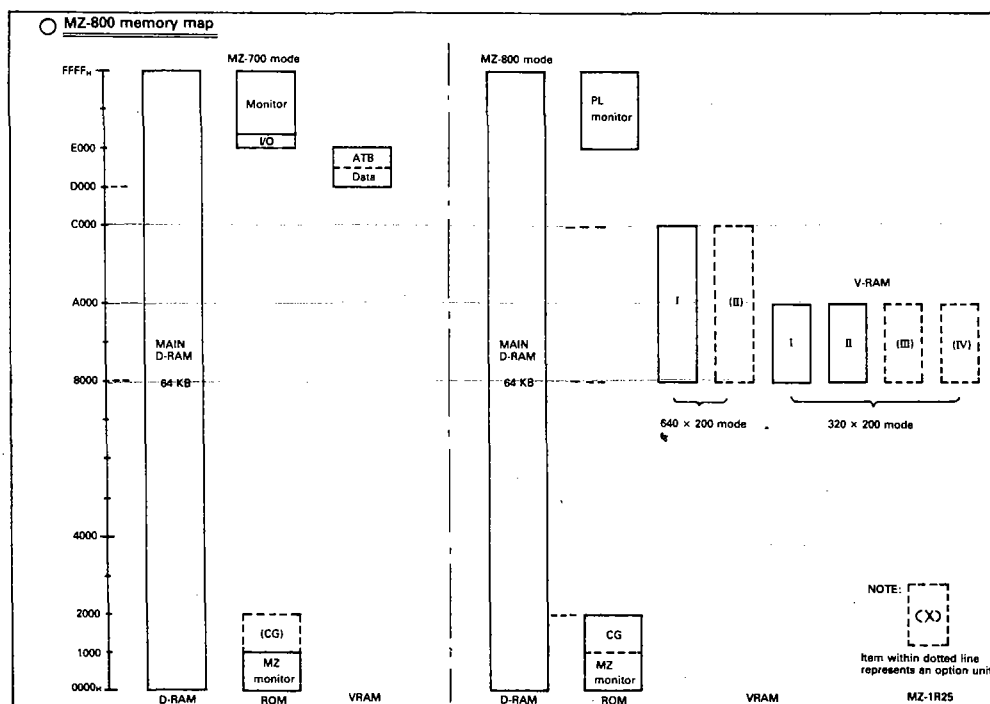




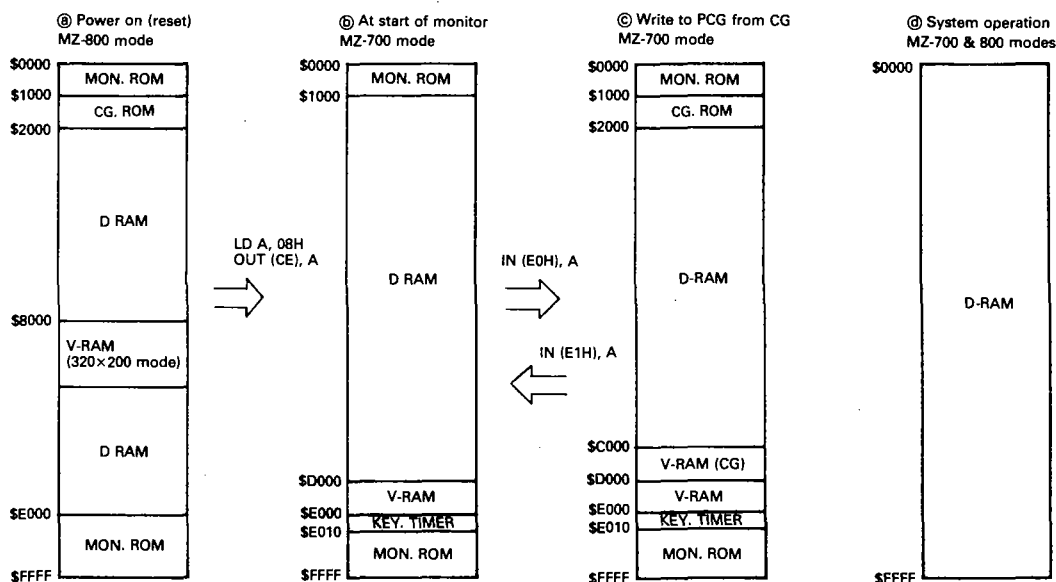
3 Memory map

The MZ-800 has a different memory map depending on

the mode. To have compatibility with the MZ-700, it has two modes of the MZ-700 mode and MZ-800 mode.



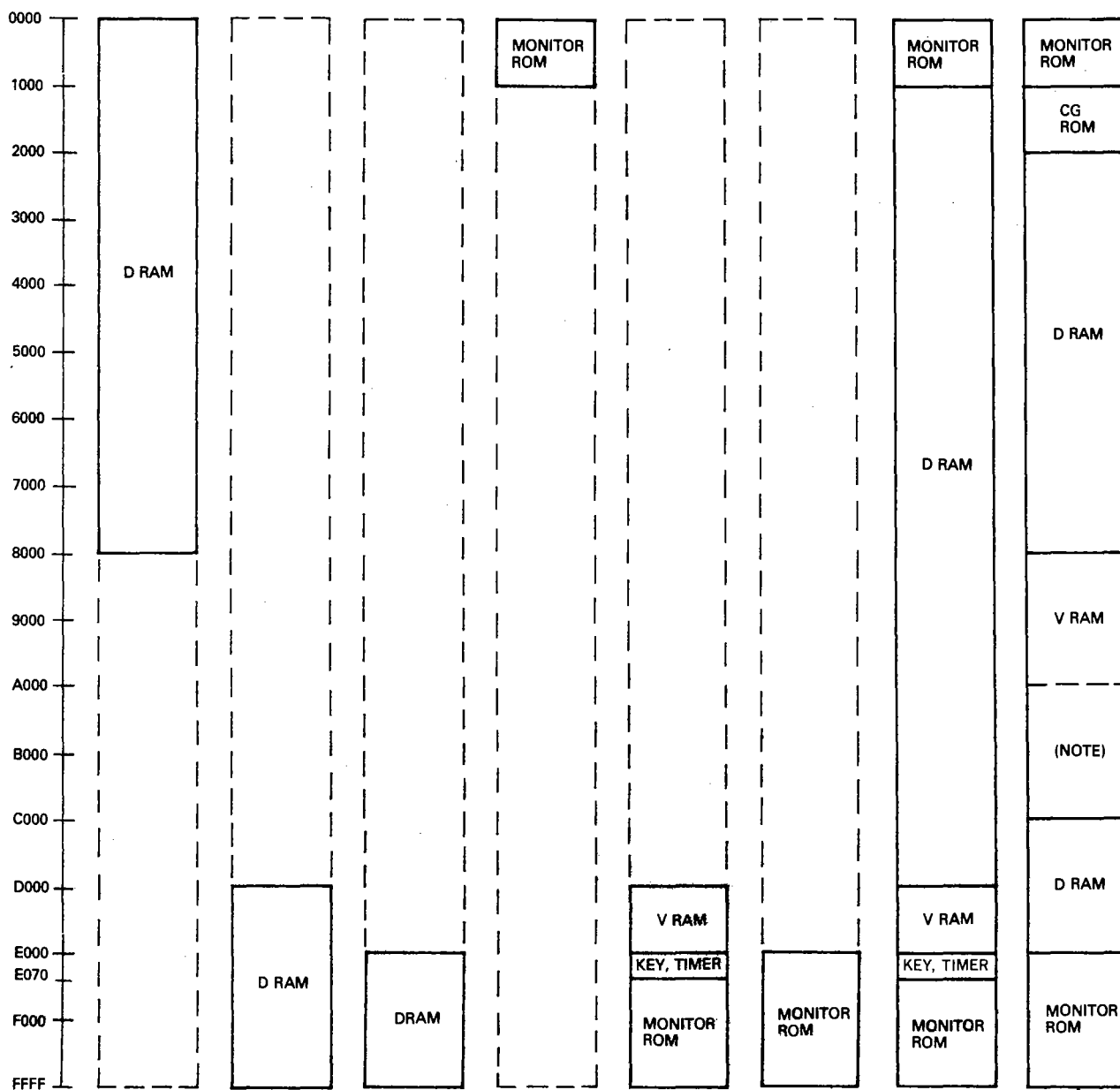
Memory map changes after initial program loading



- Memory map at power on is in the MZ-800 mode as in (a), but it changes to the MZ-700 mode by the monitor ROM when the monitor program starts. After transferring the CG data to the VRAM PCG area from the CG ROM at (c), the memory map then returns to (b).
- When the system program is completed to load, the memory map goes into the MZ-700 mode if the system switch (SW1) is set to ON side. If set to OFF side, it changes to the MZ-800 mode, then the memory map as in (d). During those changes, all memory spaces are composed of RAM and isolated from ROM and VRAM.
- Depression of the manual reset switch assumes memory map transition in order of (a) → (b) → (c) → (b), similar as in the case of power on.
- However, depression of the reset switch in conjunction with the **CTRL** key assumes the memory map of (d) after being changed once to the MZ-700 or MZ-800 mode depending on the state of the system switch. In the case of the MZ-800 mode, it is set to the plane I, II (4-color mode) of the 320 × 200 mode.

Memory Bank Control

Output port	\$E0	\$E1		\$E2	\$E3		\$E4	
MODE	—	MZ-700 mode	MZ-800 mode	—	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode
Function	◦ \$0000 ~ \$7FFF to DRAM.	◦ \$D000 ~ \$FFFF to DRAM.	◦ \$E000 ~ \$FFFF to DRAM.	◦ \$0000 ~ \$0FFF to monitor ROM.	◦ \$D000 ~ \$FFFF to VRAM, key timer, and monitor ROM.	◦ \$E000 ~ \$FFFF to monitor ROM.	◦ \$0000 ~ \$0FFF to monitor ROM. ◦ \$1000 ~ \$CFFF to DRAM ◦ \$D000 ~ \$FFFF to VRAM, key timer, and monitor ROM.	◦ \$0000 ~ \$0FFF to monitor ROM. ◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$2000 ~ \$7FFF and \$C000 ~ \$DFFF to DRAM. ◦ \$8000 ~ \$BFFF to VRAM (NOTE). ◦ \$E000 ~ \$FFFF to monitor ROM.

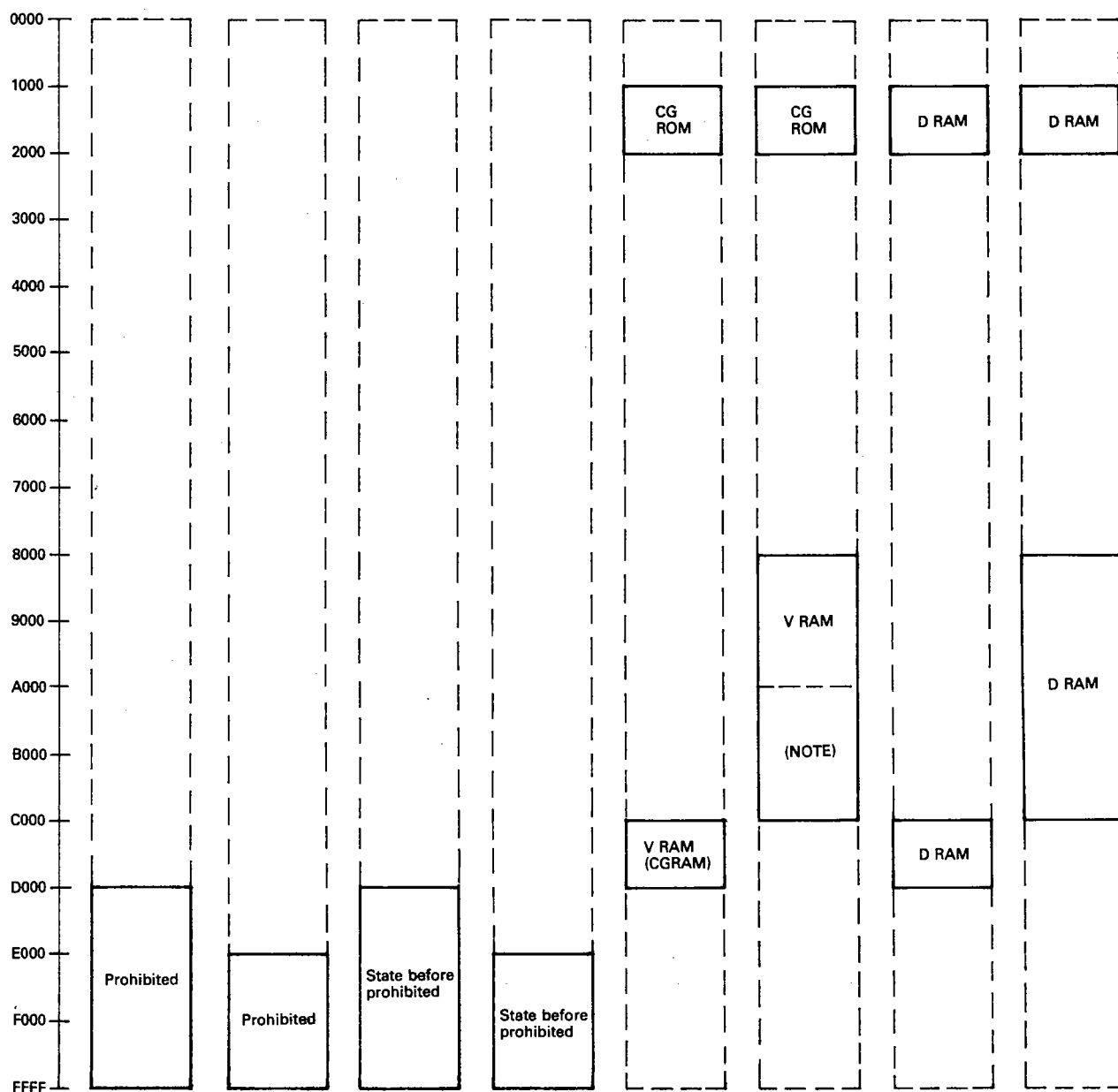


Area within dotted line does not involve change.

(NOTE): In the case of 320 × 200 mode, contents of \$8000 ~ \$9FFF are transferred, instead, and those after \$A000 are transferred to DRAM.

Power on or RESET input

IO port	OUT (\$E5)		OUT (\$E6)		IN (\$E0)		IN (\$E1)	
MODE	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode
Function	◦ \$D000 ~ \$7FFF prohibited.	◦ \$E000 ~ \$FFFF prohibited.	◦ \$D000 ~ \$FFFF returned to the state before prohibited.	◦ \$E000 ~ \$FFFF returned to the state before prohibited.	◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$C000 ~ \$CFFF to VRAM (PCG RAM).	◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$8000 ~ \$BFFF to VRAM (NOTE).	◦ \$1000 ~ \$1FFF returned to the state before CG was set. ◦ \$C000 ~ \$CFFF to DRAM.	◦ \$1000 ~ \$1FFF returned to the state before CG was set. ◦ \$8000 ~ \$BFFF to DRAM.



HARDWARE

4 Custom LSI

The custom LSI is a 100-pin single chip LSI on which the MZ-800 memory controller (I/O controller) and CRT controller, etc. are contained.

4-1 Memory controller

Used for the control of the memory bank. Addressing of DRAM, ROM, and VRAM is conducted by selection I/O address, \$E0 ~ \$E6, using OUT or IN command.

4-2 I/O controller

In this I/O controller is created the select signal for assignment of MZ-800 internal device.

See Table-2 for relation of internal device vs I/O address.

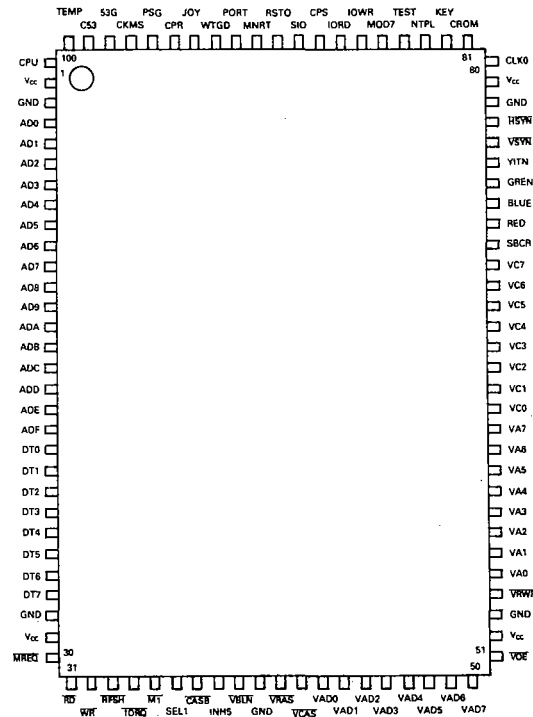
I/O address	Signal name	Device (I/O)	Function
FF FE FD FC	CPR	Z80A PIO (I/O)	Port B, printer data output Port A, printer control and timer interrupt Port B control (Mode 0) Port A control (Mode 3)
F2	PSG	PSG (O)	PSG output port
F1 F0	JOY	JOYSTICK (I)	Joystick-2 input port Joystick-1 input port
F0	—	(O)	Pallet write
E6 } E0	—	— (I/O)	Memory bank control
D7 D6 D5 D4	C53	8253 (I/O)	Control port output Counter-2 Counter-1 Counter-0 (NOTE): Mapped to E007 ~ E004 in the MZ-700 mode.
D3 D2 D1 D0	KEY	8255 (I/O)	Control Port C, cassette, etc. Port B, key input Port A, key strobe output (NOTE): Mapped to E003 ~ E000 in the MZ-700 mode.
CF CE CD CC	—	— O I/O O O	CRTC register
\$E008	—	— I/O	TEMP, HBLK input; and 8253 G0 ON/OFF output for the MZ-700 mode only.

* When above I/O address is accessed, it makes IOWR active for OUT or IORD for IN command.

Pin No.	Signal name	I/O	Functional description	Note
1	CPU	O	CPU clock (3.547 MHz)	
2	5 V	—	Power supply	
3	GND	—	Ground	
4 }	AD0 }	I	CPU address bus	
19	ADF			
20 }	DT0 }	I/O	CPU data bus	
27	DT7			
28	GND	—	Ground	
29	VCC	—	Power supply	
30	MREQ	I	CPU MREQ signal	Negative logic
31	RD	I	CPU RD signal	Negative logic
32	WR	I	CPU WR signal	Negative logic
33	RFSH	I	CPU RFSH signal	Negative logic
34	TORQ	I	CPU TORQ signal	Negative logic
35	MT	I	CPU MT signal	Negative logic
36	SEL1	O	System RAM address multiplexer select signal	
37	CASB	O	System RAM column address strobe signal	
38	INH5	O	Inhibit bank (OUT \$E5) select signal ("H" = Inhibit).	OPEN
39	VBLN	O	Vertical blanking signal	Negative logic
40	GND	—		
41	VRAS	O	VRAM RAS control signal	Negative logic
42	VCAS	O	VRAM CAS control signal	Negative logic
43 }	VAD0 }	O	VRAM address signal (multiplexer output)	
50	VAD7			
51	VOE	O	VRAM output enable	Negative logic
52	VCC	—	Power supply	
53	GND	—	Ground	
54	VRWR	O	VRAM write signal	Negative logic
55 }	VA0 }	I/O	VRAM data bus (standard RAM)	
62	VA7			
63 }	VC0 }	I/O	VRAM data bus (option RAM)	
70	VC7			
71	SBCR	O	Color sub-carrier wave	
72	RED	O	Video signal, red	
73	BLUE	O	Video signal, blue	
74	GREEN	O	Video signal, green	
75	YITN	O	Brightness control signal	
76	VSYN	O	Vertical sync signal	Negative logic
77	HSYN	O	Horizontal sync signal	Negative logic
78	GND	—		
79	VCC	—		
80	CLK0	I	Clock input (17.7344 MHz)	
81	CROM	O	ROM chip enable	Negative logic
82	KEY	O	8255 chip enable	Negative logic
83	NTPL	I	NTSC/PAL selection (PAL = "L")	GND
84	TEST	I	Test pin ("H" = test mode)	GND
85	MOD7	I	MZ-700/800 mode selection ("L" = MZ-700 mode)	
86	TOWR	O	Sum of CS and WR of I/O controlled by the custom IC	Negative logic
87	TORD	O	Sum of CS and RD of I/O controlled by the custom IC	Negative logic
88	CRS	O	I/O \$B0 ~ \$B3 chip enable	OPEN
89	SIO	O	I/O \$F4 ~ \$F7 chip enable	OPEN
90	RSTO	O	Reset output	Negative logic
91	MNRT	I	Manual reset input	Negative logic
92	PORT	I	Power on reset input	Negative logic
93	WTGD	O	Wait signal to CPU	Open drain
94	JOY	O	Joystick chip enable	Negative logic
95	CPR	O	PIO chip select	Negative logic
96	PSG	O	76489 chip select	Negative logic
97	CKMS	O	8253 musical interval clock	
98	53G	O	8253 musical interval ON/OFF gate signal	
99	C83	O	8253 chip enable	Negative logic
100	TEMP	I	MZ-700 mode, \$E800 tempo input	

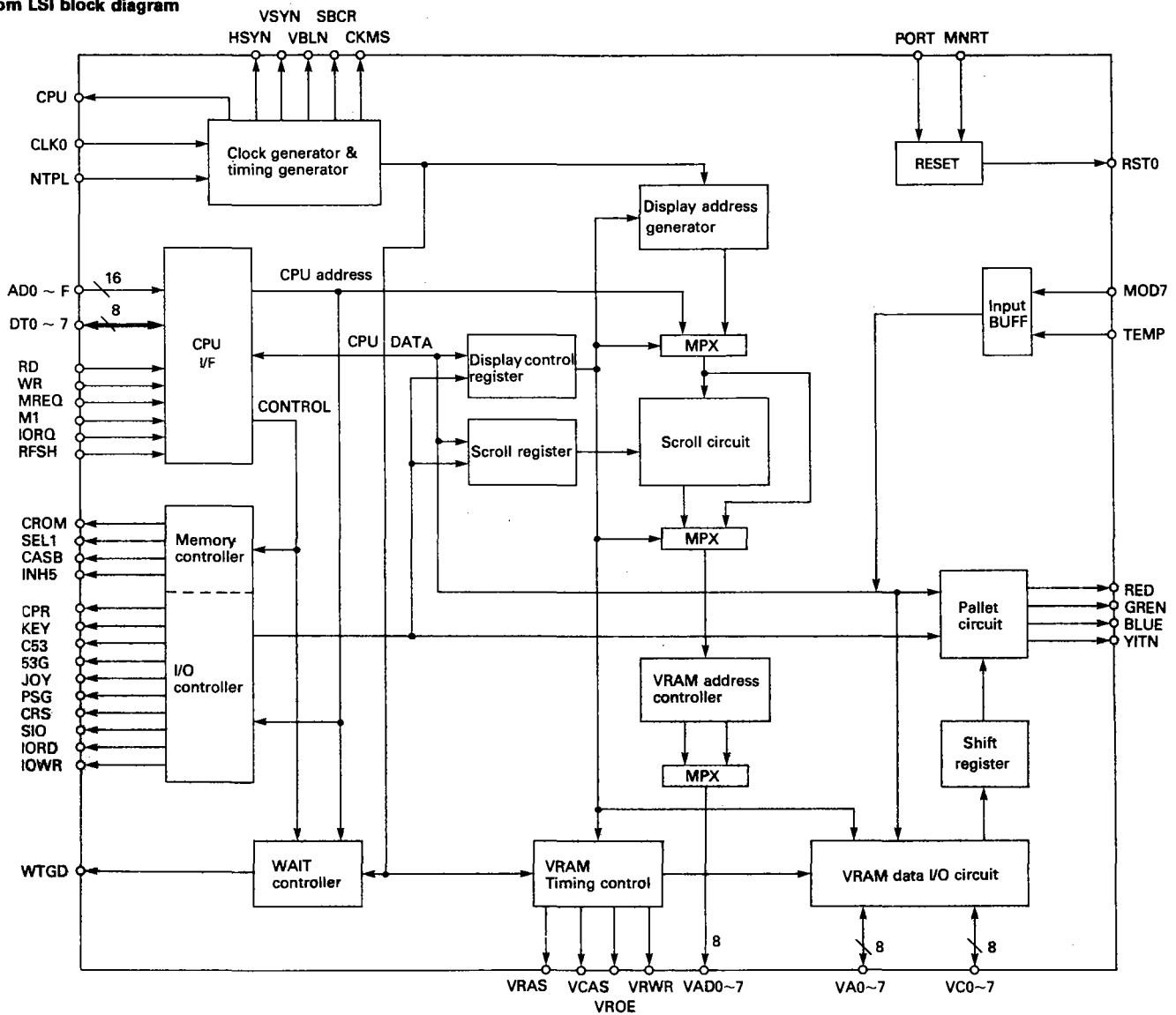
* Term "OPEN" represents the signal not used on the board.

Pin configuration



V_{CC} 2, 29, 52, 79, pin
GND 3, 28, 40, 53, 78 pin

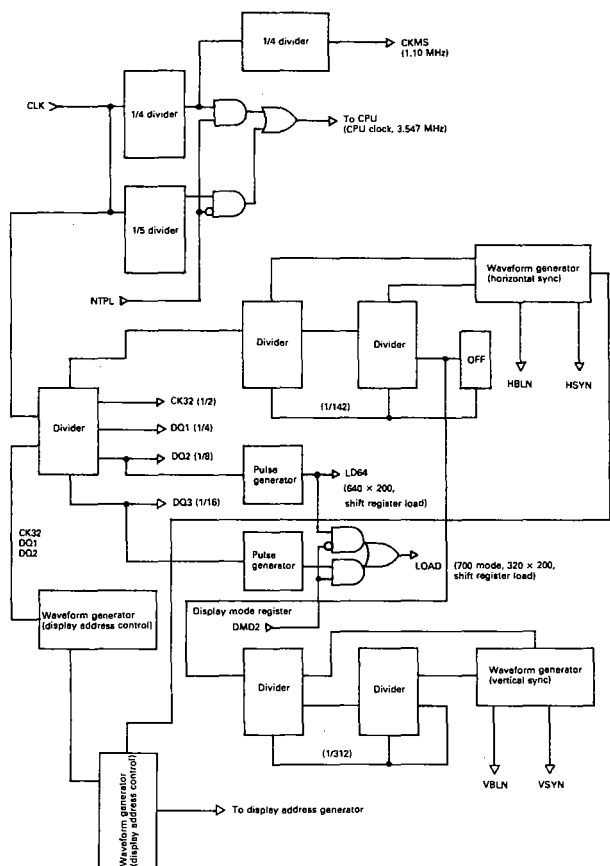
Custom LSI block diagram



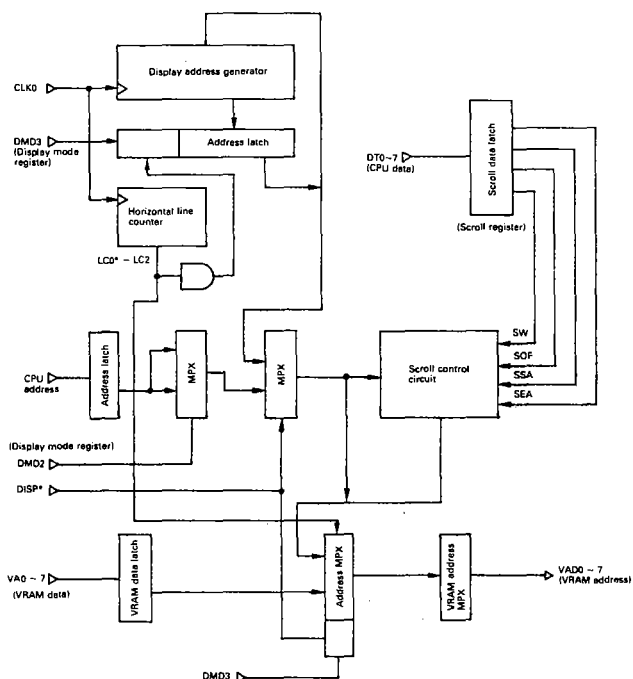
4-3 Clock generator and timing generator

Oscillation from the crystal oscillator is divided to create the CPU clock, horizontal sync, vertical sync, and display address control signals.

Since the low state of signal is used for NTPL (NTSC/PAL selection) with the MZ-800, the CPU clock of 3.547 MHz is derived from the crystal frequency of 17.734 MHz by dividing it 1/5.



Clock generator and timing generator circuits



Display address generator block diagram

4-4 Display address generator

1) Display address generation

- Display address increments from left to right as beginning from the home position at the upper left corner of the CRT screen (address \$000). The first display line dominates address \$000 through \$027. Because a screen frame consists of 200 rasters, the address at the right side of the bottom corner is as follows:

$$(200 \times 40) - 1 = 7999 = \$1F3F$$

- The address counter stops counting for a horizontal flyback line and stored in the address latch circuit. When the horizontal flyback line terminates, the address latch output is preset in the address counter (display address generator).
- Address is generated even while the vertical flyback line is active and it makes the counter reset before termination of the vertical flyback line.

2) Display address generation in the MZ-700 mode

- Because characters are displayed under the PCG method in the MZ-700 mode, address is generated for each character and the same address is used for displaying of one character. The 3-bit horizontal line counter is provided to count horizontal lines to generate the address (LC0 ~ LC2) for selection of the character front.

Display address increments from left to right having the uppermost left corner of the screen for the home position.

Since 25 lines are used to develop displaying of characters composed of 8×8 dots, the address at the right of the bottom lines becomes \$3EF.

3) Display address multiplexed with CPU address

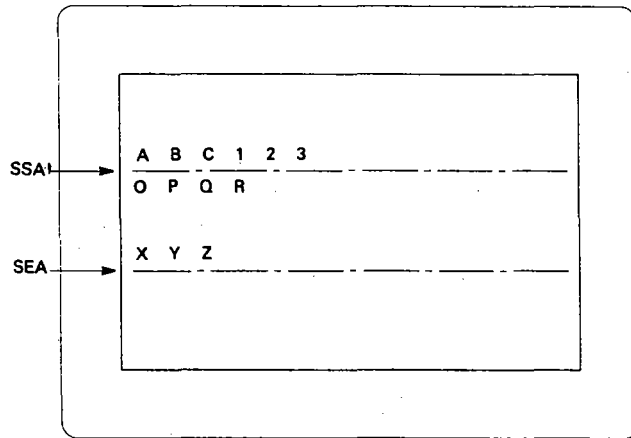
- Address used to write data to the VRAM is latched in order to avoid CPU wait. Display modes of 640 dots and 320 dots are assigned by the mode switch (DMD2).
- Display address is multiplexed with the VRAM write address in the timing of DISP which has the timing that the display address and CPU address may become a pseudo cycle steal.

4-5 Scroll

- 1) Scrolling is possible for both horizontal and vertical directions by means of software offset.

The following four registers are used for scroll control.

- a. Scroll start address register: SSA (7-bit)
- b. Scroll end address register: SEA (7-bit)
- c. Scroll width register: SW = SEA-SSA (7-bit)
- d. Scroll offset register: SOF (10-bit)



- 2) Control of scroll starts by the initialization of the scroll control register.

SSA = \$0
SEA = \$7D
SW = \$7D
SOF = \$0

- 3) Way of smooth scrolling

SOF = \$0 → \$5

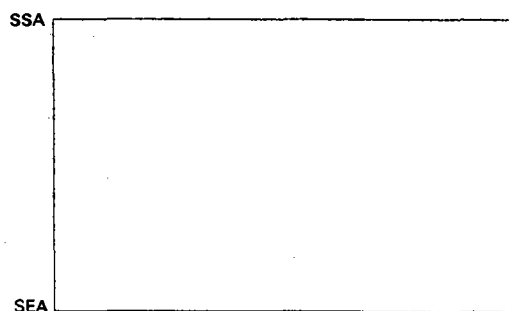
Programming "SOF = \$5" makes the display screen shifted one line up.

The highest line (address: \$0 ~ \$27) is then assigned to the lowest line (\$1F18 ~ \$1F3F).

As normal scroll involves updating of the data for the lowest line, the data of address \$1F18 ~ \$1F3F are updated.

SOF = \$5 → \$0

By reducing the value of SOF by "5", it makes the screen shifted one line down.



- 4) Line scroll

SOF = \$0 → \$28

Programming "SOF = \$28" makes the display screen shifted eight lines up. Data on the highest line therefore shifted to the bottom line.

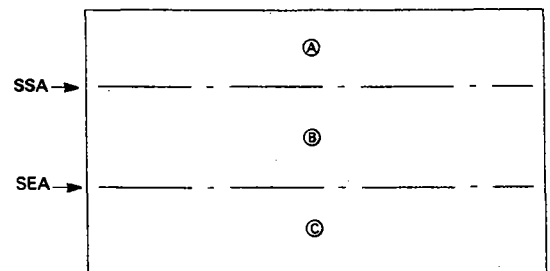
Programming "\$28 → \$0" makes the display screen shifted eight lines down, and the line on the bottom moves to the highest line.

- 5) Screen split

Appropriate deviation of SSA, SEA, and SW permits to divide the screen into three sections of ①, ② and ③.

Though the section ② is permitted to scroll, sections ① and ③ are not permitted to scroll.

See the figure to explain with.



Assume now that the top of the section ② is on the 5th line (40 raster) and the top of the section ③ is on the 18th line (144 raster). Attention must be paid to the fact that values SSA and SEA are used for assigning lines. Scroll registers are set with the following values.

SSA = \$19
SEA = \$5A
SW = \$41
SOF = \$0

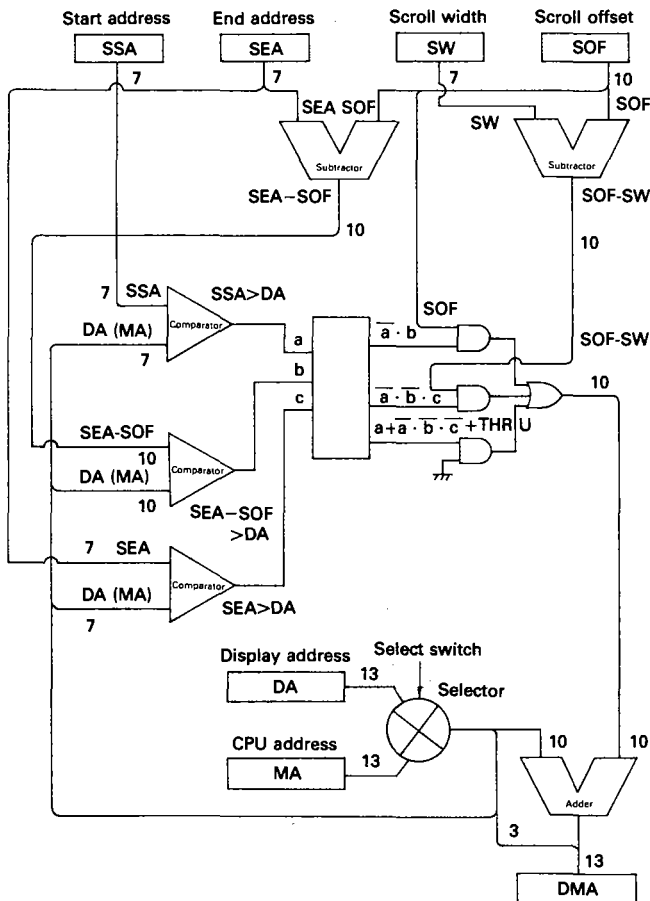
In this occasion, it needs to initialize the screen that has been displayed. "SOF = \$5" must be programmed to scroll ② one line. Then, only the section ② is shifted up, and the highest line of ② moves to the bottom line of ②. Programming "SOF = \$A" makes it scrolled one more line.

SOF ≤ SW

Scroll offset (SOF) should necessarily be within a range of the scroll width. Display is not assured with SOF set greater than SW.

Scroll and control circuit hardware

• Block diagram



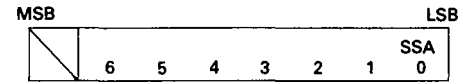
Scroll control register

SSA: Scroll start address

Increment of SSA: \$5

Minimum value of SSA: \$0

Maximum value of SSA: \$78

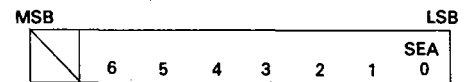


SEA: Scroll end address

Increment of SEA: \$5

Minimum value of SEA: \$5

Maximum value of SEA: \$7D



SW: Scroll width

Increment of SW: \$5

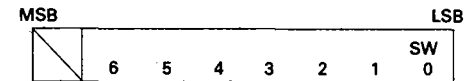
Minimum value of SW: \$5

Maximum value of SW: \$7D

Relation of SW, SEA, vs SSA

$$SW = SEA - SSA$$

$$SW > SSA$$

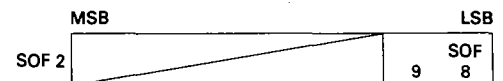
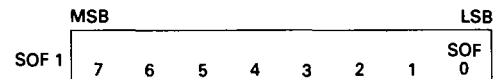


SOF: Scroll offset

Increment of SOF: \$5

Minimum value of SOF: \$0 (without offset)

Maximum value of SOF: \$3E8



Relation of display address, SEA, SSA, vs SOF

Display address	m	l	k	j	i	h	g	f	e	d	c	b	a	
SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	
SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	
SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	
Screen left end address														
0 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	
1 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	
2 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	
3 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	First line
8 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	
16 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	Second line
24 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	
192 Line	1	1	1	1	0	0	0	0	0	0	0	0	0	Twenty fifth line
199 Line	1	1	1	1	0	0	0	0	0	0	0	0	0	

Relation of SW vs SOF

$$SW > SOF$$

Concept of the scroll control circuit

Scroll method

- Scrolling by means of VRAM address conversion.

Range of scroll

- y-axis programmable.
BASIC console command compatible
- x-axis fixed

Scroll sequence

- The scroll start address is termed "SSA" and end address "SEA".
- Execution of scroll, with offset given from the CPU.
- One line (line S) starting from SSA disappears from the display screen.
- A new line (line S') is added to SEA. Line S' is the same refresh memory as the line S. The contents of the memory was erased (nullified by the CPU) before the execution.

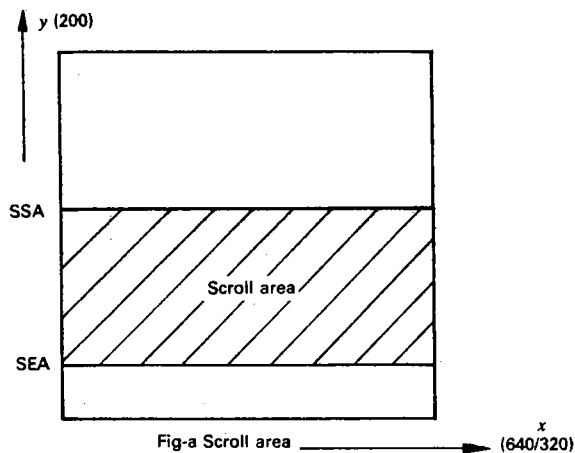


Fig-a Scroll area

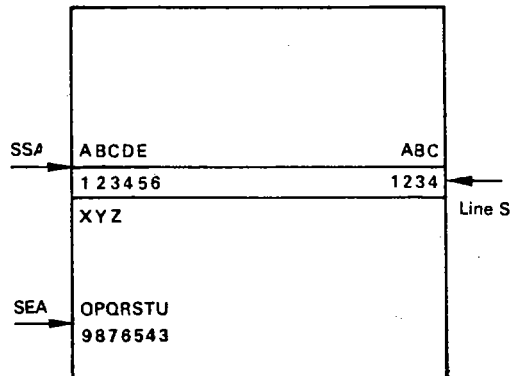


Fig-b Screen before scroll

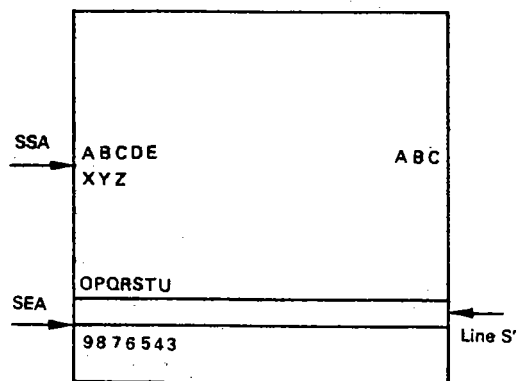
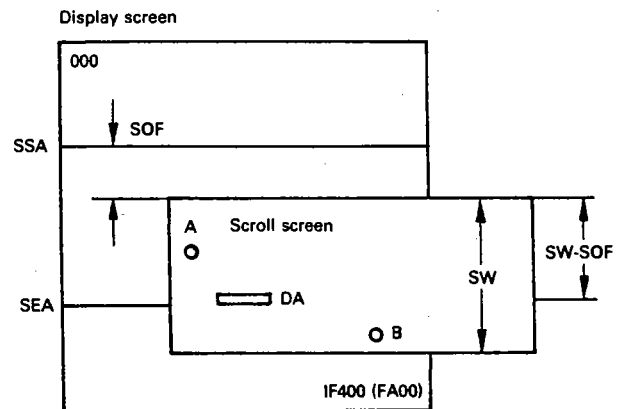


Fig-c Line after scroll

Execution of scrolling by address conversion

- Scroll offset (SOF) is the count of lines which the CPU gives to the CRTC. For instance, the following must be observed to perform scrolling.
3-line scroll: $SOF_3 = 0F \times 3$
5-line scroll: $SOF_5 = 0F \times 5$
And, to scroll one more line after 5-line scroll;
5-line scroll: $SOF_5' = SOF_5 + 0F = 0F \times 6$



- Display address DA is the signal created in the CRTC display address generation circuit and arranged in their order from the upper left corner of the screen. The bottom right address is 1F400 in the 640 × 200 mode.
- Display memory address DMA represents the VRAM address corresponding to DA. Since scroll is executed by means of address conversion, the order of DMA may not be the same as DA, necessarily.
- CPU address MA is the VRAM address that obtained from the CPU through the CRTC. To lighten burden on the CPU, a circuit is added to make order of DA identical to order of MA arrangement.

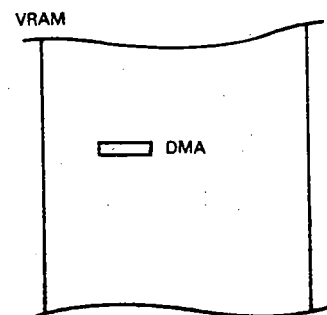


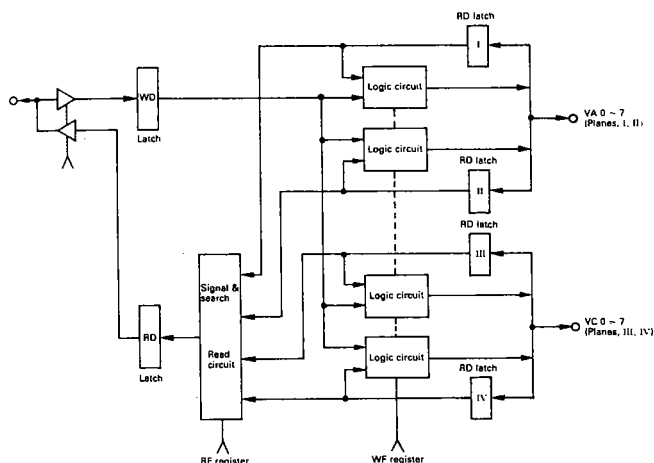
Fig-d Address conversion

4-6 VRAM data input/output circuit

1. Nothing intervenes for input and output of data in the case of the MZ-700 mode.
2. MZ-800 mode
 - Write
Read data (RD) from the VRAM and write data (WD) from the CPU are subjected to logical operation according to the direction from the write format register (WF) and its result is written.

- Read

For plane read data from the VRAM, data to be read by the CPU are arranged in accordance with the direction of the read format register (RF).



* Logic circuit

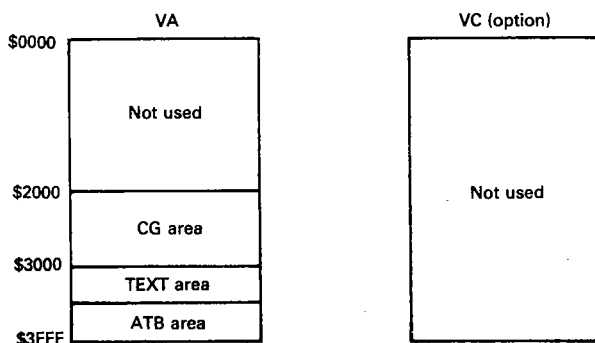
Read data from the VRAM and write data from the CPU are subjected to logical operation (OR, XOR, RESET, etc.) and its result is used for the write data.

VRAM access timing

1) MZ-700 mode

See separate page for display timing chart.

The VRAM is configured in the following manner in this instance.



As the PCG method is adopted for the MZ-700 mode, the text and ATB areas are actually mapped to \$D000 ~ \$DFFF. So, the VRAM address has the following relation with the display character position.

	1	2	3	...	40
1	D000	D001	D002	...	D027
2	D028			...	
...				...	
25	D3C0			...	D3E7

2) MZ-800 mode

As the bit map method is used for the MZ-800 mode, it is possible to four screens of 320×200 dots and two screens (maximum) of 640×200 dots.

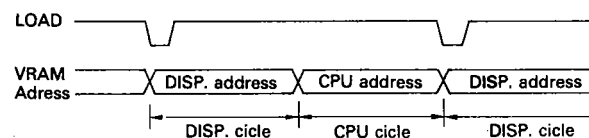
The cycle steal method is used for this mode.

i) 320×200 dots

See separate page for the timing chart during display and CPU read timing.

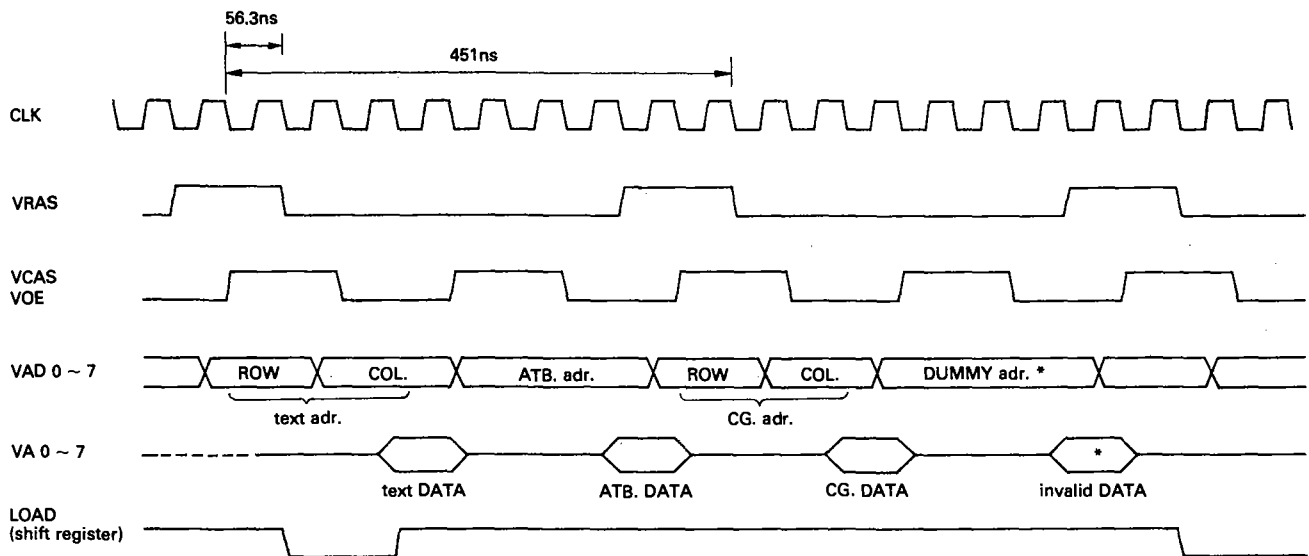
What is pseudo cycle steal

With the MZ-800, the pseudo cycle steal method is adopted for VRAM accessing.

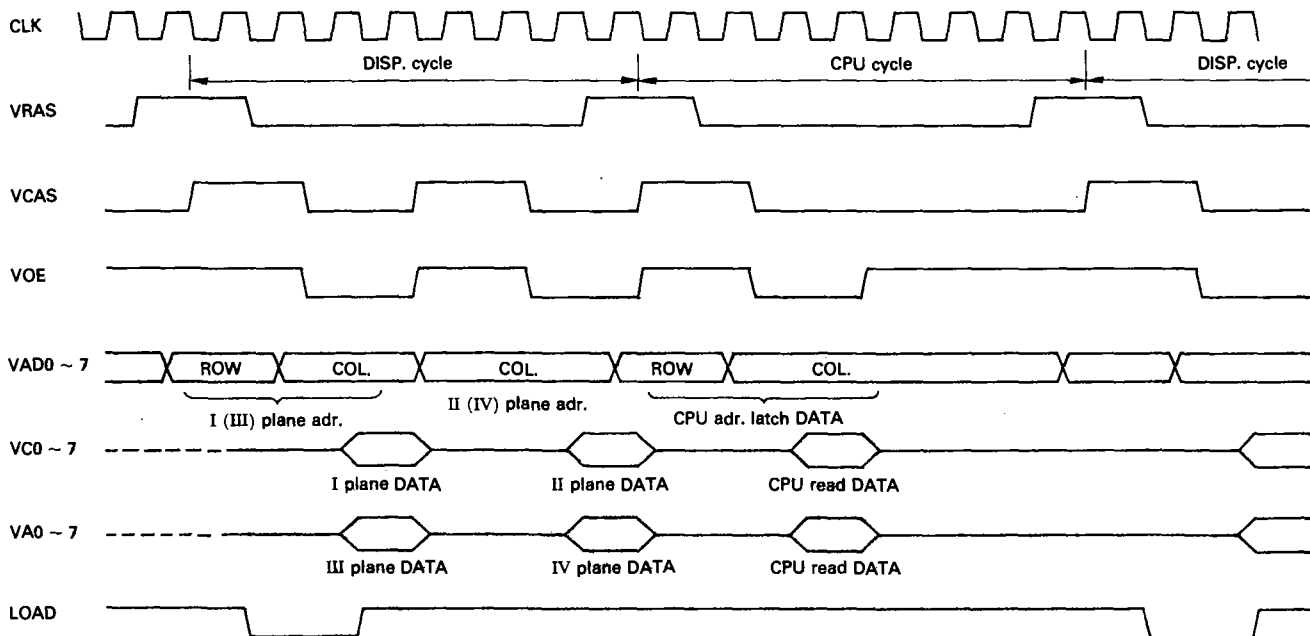


As shown in the figure, a next display data fetch and CPU accessing are multiplexed during a display period. Because accessing of the VRAM while characters are on display causes the screen to blink with the MZ-700 mode, it awaits for blinking to complete before accessing of the VRAM. But, with the cycle steal method it enhances faster screen processing as it enables to access the VRAM during a display period. Because it is not a complete cycle steal with the MZ-800 but timing is taken using a wait in order to synchronize with the CPU cycle for accessing from the CPU, it is therefore called "pseudo cycle steal".

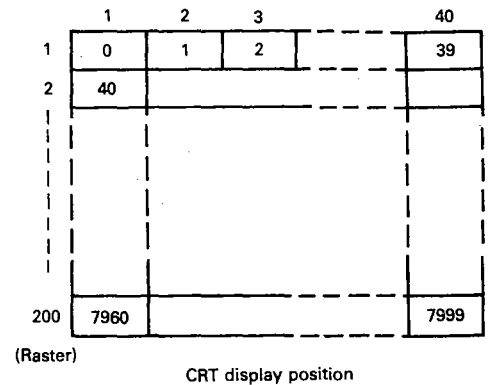
MZ-700 MODE DISPLAY TIMING



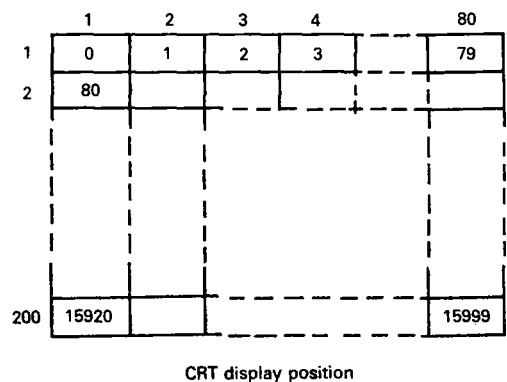
MZ-800 MODE (320 × 200 dot)



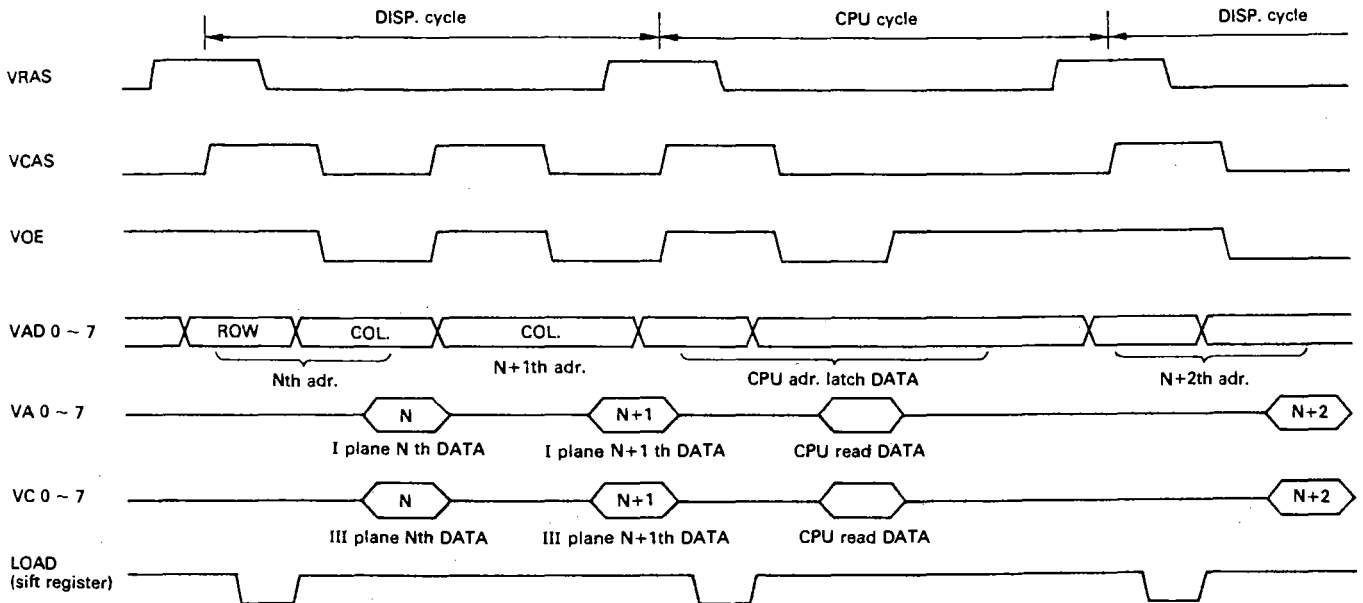
See the figure below for VRAM configuration and CRT character display position.



Because it operates in the cycle steal mode, two bytes of display data are fetched during one byte display cycle. (See the chart in separate page.) See the figure below for VRAM configuration and CRT character display position.



800 MODE (640 × 200 dot)



CPU and VRAM accessing

1. Accessing of the VRAM by the CPU is carried out in the cycle steal mode (MZ-800 mode only) during the flyback period of the display under the control of the CRT controller.
2. Even when there is no accessing from the CPU in the CPU cycle, such as VRAS, VCAS, VOE, etc. are outputted in the timing of the read cycle at all times.
3. Write to the VRAM is carried out after logical operation of the read and write data by means of the read-modify-write method. But, in the case of the 320 × 200, 16-color mode, data are written in two CPU cycles as there is a need of writing to Plane IV. See separate paper for timing chart.

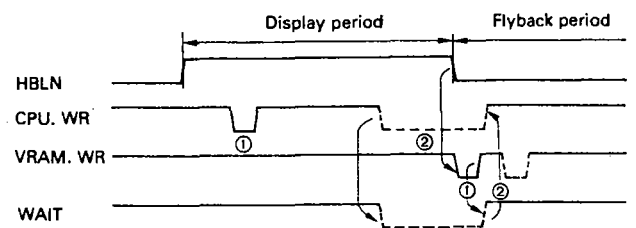
4. CPU wait

1) Write

- As there is a one-byte buffer in the CRT controller, write to the VRAM from the CPU is carried out through the buffer. But, actual write to the VRAM is

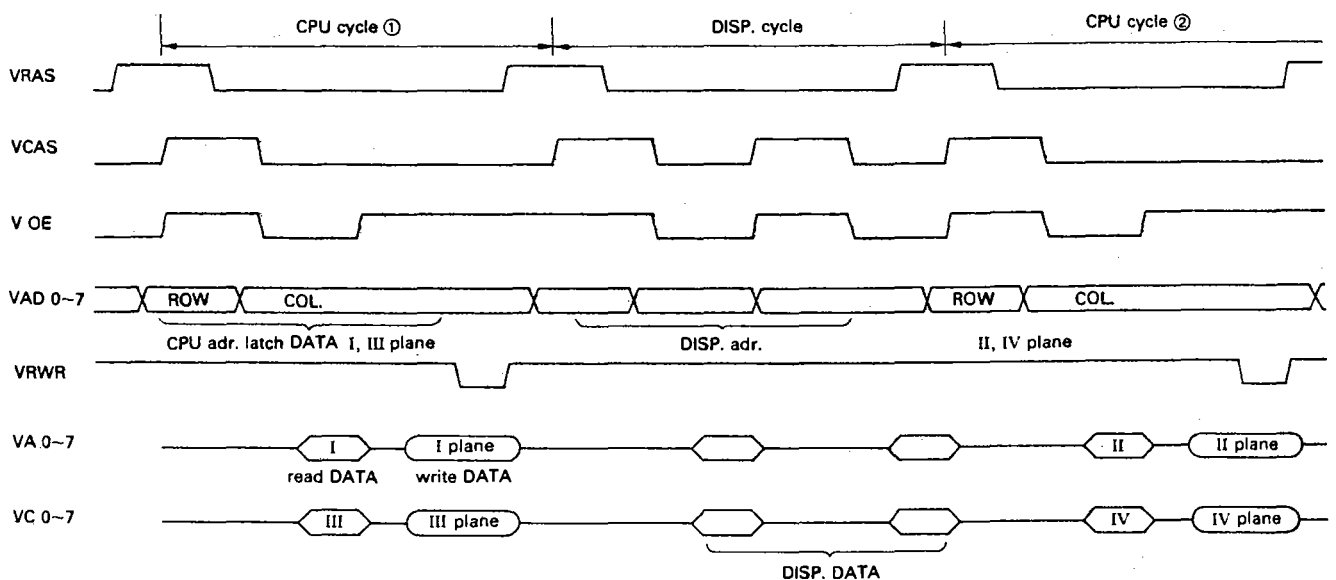
done by the CRT controller. Therefore, there would be no need of wait under almost any condition in the MZ-800 mode.

- Even in the MZ-700 mode, wait is issued when there are more than two writes in a display period.



2) Read

Wait is issued along with the CPU write action both during displaying and flyback periods to perform reading operation in synchronization with the CPU cycle.



4-7 Register functions

VRAM configuration

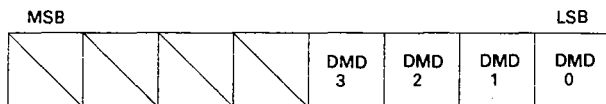
- One or two chips of 16 KB VRAM are used.
- In the case of a single 16 KB VRAM chip, it handles 320 × 200 dots, 4 colors, or 640 × 200 dots 1 color.
- In the case of two 16 KB VRAM chips, it handles 320 × 200 dots, 16 colors, 640 × 200 dots, 4 colors, 320 × 200 dots, 4 colors, 2 frames, or 640 × 200 dots, 1 color, two frames.

* Discussed next are about functions of the custom LSI.
There may be some restrictions because the standard version of the MZ-800 incorporates only one 16 KB RAM.

Display mode register (OUT &HCE)

- It consists of four bits which are used to represent display method, resolution, and display screen (color plane) in combined way.

Display mode register (DMD)



DMD 3, 2: Display method and resolution

DMD 3	2	
0	0	Bit map, 320 × 200
0	1	Bit map, 640 × 200
1	0	MZ-700 mode
1	1	Prohibited

DMD 1, 0: Display screen designation

DMD 1	DMD 0	320 × 200	640 × 200	MZ-700
0	0	Frame A, Planes I and II	Frame A, Plane I	Normal
0	1	Frame B, Planes III and IV	Frame B, Plane III (NOTE)	Prohibited
1	0	Planes I, II, III, and IV	Planes I, III	Prohibited
1	1	Prohibited		

NOTE: 640 × 200, Plane B is Plane III, not Plane II.

* With the MZ-800, DMD 1 = 0, DMD 0 = 0.

△ 3/3

Table-1 VRAM configuration and display mode

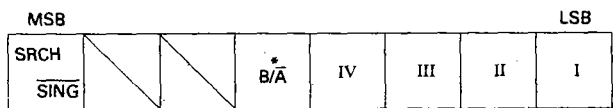
VRAM capacity	VRAM configuration	Resolution	Display color	Display frame	Color combination (NOTE)	DMD			
						3	2	1	0
16 KB		320 × 200	4 colors	Frame A	I, II	0	0	0	0
		640 × 200	1 color	Frame A	I	0	1	0	0
32 KB		320 × 200	4 colors	Frame A	I, II	0	0	0	0
				Frame B	III, IV	0	0	0	1
			16 colors	Frame A	I, II, III, IV	0	0	1	0
		640 × 200	1 color	Frame A	I	0	1	0	0
				Frame B	III	0	1	0	1
			4 colors	Frame A	I, III	0	1	1	0
	MZ-700	40 characters × 25 lines	8 colors	Frame A	RGB	1	0	0	0

(NOTE) Except for the MZ-700 mode, actual display colors are produced by the pallet.

VRAM to CPU interface

- As the CRTC bus is completely separated from the CPU bus, read and write of the VRAM is carried out through the CRTC. Therefore, interfacing with the CPU is done via the read register or write register in the CRTC.
- VRAM access by the CRTC is done under the pseudo cycle steal mode.
- Not only read and write are for the accessing with the CPU, it permits to read multiple number of screen data logical operational results and to write the read-modify-write of the logical operational results for the data already written. So, it has two registers of the read format register and the write format register.
- It permits CPU access to the non-display plane in the display mode according to the $B/\bar{A}$ bit and it enables selection of data buffer and two screens, when the 32 KB VRAM is used.

a) Read format register (RF) (OUT & CD)



* NOTE: Same as the bit $B/\bar{A}$ of the write format register.

• $\text{SRCH}/\overline{\text{SING}}$

"0": Single color data read

Reads the data of the color plane, I, II, III, or IV, specified by "1".

NOTE: Only one item should be "1" out of I, II, III, and IV. If it is "1" for more than two or non-existence of the VRAM may not assure the data read.

"1": Specified color search

"1" is returned for the bit of the color specified by 0/1 of I, II, III, and IV.

NOTE: Depending on the display more, color combination is permitted for the bit combination of I, II, III, IV; III, IV; I, II; I; and III. Bit combination otherwise will be disregarded.

(ex. For the 640×200 , 4-color mode, combination becomes possible for I and III, and II and IV are disregarded.

• $B/\bar{A}$

CPU access plane change

MZ-800 $\rightarrow$ "0": Frame A access

Accesses the frame A (planes I and II for the 320×200 , 4-color mode; plane I for the 640×200 , 1-color mode).

"1": Frame B access

Accesses the (planes III and IV for the 320×200 , 4-color mode; plane II for the 640×200 , 1-color mode).

• I, II, III, IV Color plane designation.

Table-2 Display mode vs read format register

	Display mode	SRCH/SING	B/A	IV	III	II	I	Function (NOTE)
Single color data read	320 × 200, 4/16 colors	0	(*) Frame A: "0" Frame B: "1"	0	0	0	1	Plane I data read
				0	0	1	0	Plane II data read
				0	1	0	0	Plane III data read
				1	0	0	0	Plane IV data read
Specified color search	320 × 200, 4 colors	1	0	×	×	0	0	I, II dot search
				×	×	0	1	I, II dot search
				×	×	1	0	I, II dot search
				×	×	1	1	I, II dot search
			1	0	0	×	×	III, IV dot search
				0	1	×	×	III, IV dot search
				1	0	×	×	III, IV dot search
				1	1	×	×	III, IV dot search
	320 × 200, 16 colors	1	×	0	0	0	0	I, II, III, IV, dot search
				0	0	0	1	I, II, III, IV, dot search
				0	0	1	0	I, II, III, IV, dot search
				0	0	1	1	I, II, III, IV, dot search
				0	1	0	0	I, II, III, IV, dot search
				⋮	⋮	⋮	⋮	⋮
				1	1	1	1	I, II, III, IV, dot search
	640 × 200, 1 color	1	0	×	×	×	0	I, dot search
				×	×	×	1	I, dot search
			1	×	0	×	×	III, dot search
				×	1	×	×	III, dot search
	640 × 200, 4 colors	1	×	×	0	×	0	I, III, dot search
				×	0	×	1	I, III, dot search
				×	1	×	0	I, III, dot search
				×	1	×	1	I, III, dot search
	MZ-700	0	0	0	0	0	1	Data, ATB, CG area read

(*) : Refer to the display frame of Table-1.

NOTES:

- Read for the non-existing VRAM are not assured.
- The above parameter has to be set up for the MZ-700 mode.
- ★ B/A must be set to "0" for the standard MZ-800 (without MZ1R25).

b) Write format register (WR) (OUT & CC)

MSB							LSB
WMD 2	WMD 1	WMD 0	(NOTE) B/A	IV	III	II	I

NOTE: Same as the bit B/A of the read format register.

- I, II, III, IV
Color plane designation

• WMD 0 ~ 2

Selects the logical operational mode for read-modify-write.

• B/A (NOTE)

Standard MZ-800 → "0": Frame A access

Frame A is accessed for the display mode.

"1": Frame B access

Frame B is accessed for the display mode.

Write mode	WMD			B/A	Color plane				Display mode	Function	WD: Write data VD: VRAM data
	2	1	0		IV	III	II	I			
SINGLE WRITE	0	0	0	(*)	0/1	0/1	0/1	0/1	320 × 200, 4/16 colors 640 × 200, 1/4 colors	Color plane of "1": WD, write Color plane of "0": Fixed	
EXOR	0	0	1	Frame A: 0	0/1	0/1	0/1	0/1		Color plane of "1": $WD \oplus VD$ Color plane of "0": Fixed	
OR	0	1	0	Frame B: 1	0/1	0/1	0/1	0/1		Color plane of "1": $WD + VD$ Color plane of "0": Fixed	
RESET	0	1	1		0/1	0/1	0/1	0/1		Color plane of "1": $\overline{WD} \cdot VD$ Color plane of "0": Fixed	
REPLACE	1	0	X	0	X	X	0/1	0/1	320 × 200 (A), 4 colors (B)	Writes WD in a specific color (Character write to the graphic plane)	
				1	0/1	0/1	X	X			
				X	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": WD	
				0	X	X	X	0/1	640 × 200 (A), 1 color (B)	Color plane of "0": Writes "0". Color plane of "X": Fixed	
				1	X	0/1	X	X			
				X	X	0/1	X	0/1	640 × 200, 4 colors		
PSET	1	1	X	0	X	X	0/1	0/1	320 × 200 (A), 4 colors (B)	Writes only bit "1" of WD in a specific color. (Character write to graphic plane)	
				1	0/1	0/1	X	X			
				X	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": WD + VD	
				0	X	X	X	0/1	640 × 200 (A), 1 color, (B)	Color plane of "0": $\overline{WD} \cdot VD$ Color plane of "X": Fixed	
				1	X	0/1	X	X			
				X	X	0/1	X	0/1	640 × 200, 4 colors		
MZ-700	0	0	0	0	0	0	0	1	MZ-700	Writes WD into the DATA, ATB, and CG area.	

(*) Refer to Table-1 display frame

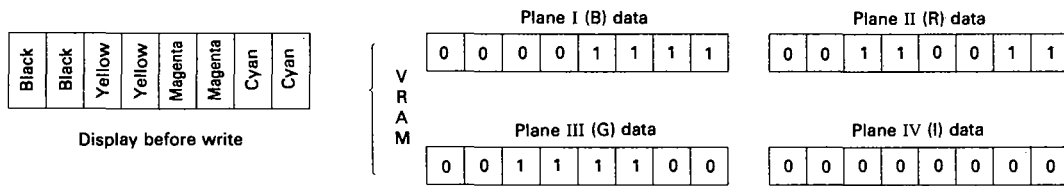
NOTES:

- Write for the non-existing VRAM are not assured.
- The above parameter has to be set up for the MZ-700 mode.
- B/A must be set to "0" for the standard version MZ-800.

c) Example of CPU read/write access

- Shown next are access examples of REPLACE write, PSET write, and SEARCH read in the 320×200 , 16-color mode.

As for display colors, Plane I corresponds to B, II to R, III to G, and IV to I.

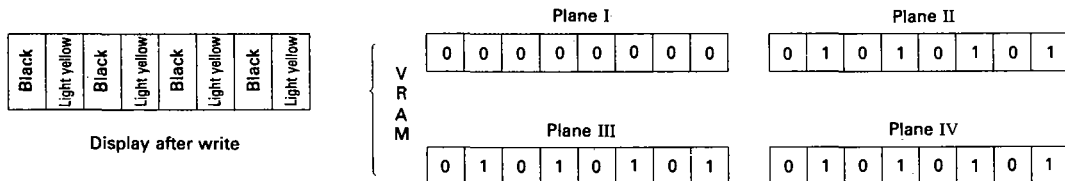


It develops the screen when a next CG patterns are written after setting the REPLACE mode and the light yellow color in the WF register.

So, the bit "1" of the write data becomes the color specified by WF and rest of others become RESET (black).

	Mode			B/A	Color designation					Write data (CG pattern)								
WF register	1	0	0	0	1	1	1	0	: Light yellow replace mode		0	1	0	1	0	1	0	1

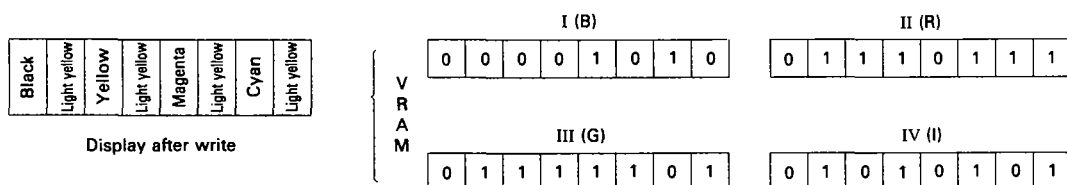
: Light yellow replace mode



② PSET write

- To overlay a light yellow hatching over the graphic display screen of ①.

	Mode				B/ $\bar{A}$	Color designation											
WF register	1	1	0	0	1	1	1	0	: Light yellow PSET mode								
					I	G	R	B									
Write data	<table><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>								0	1	0	1	0	1	0	1	
0	1	0	1	0	1	0	1										



So, only the bit "1" of the write data becomes the color specified by WF in this mode, and rest of other colors do not change.

Read data 0 1 0 1 0 1 0 1 : Only the bit of light yellow becomes "1".

③ SEARCH read + PSET write

- To change light yellow in ② above to change to red
- The following data are set when the memory is read after setting the light yellow search mode in the RF register.

	Mode	B/A Color designation						
RF register	1		0	1	1	1	0	: Light yellow search

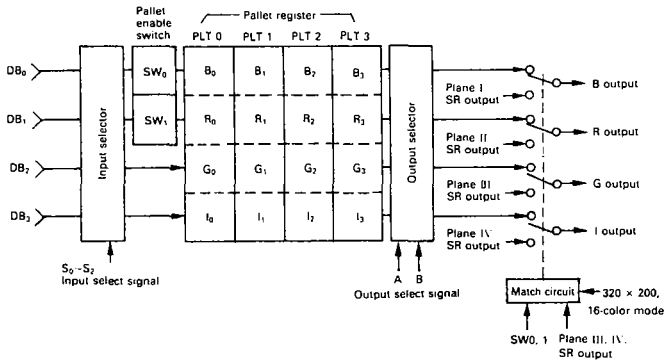
Black	Red	Yellow	Red	Magenta	Red	Cyan	Red
-------	-----	--------	-----	---------	-----	------	-----

Now, a partial color change has been attained. As in above, it enhances fast display change with less of VRAM accessing by using various write modes.

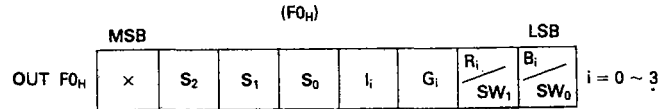
4-8 Pallet

- As there are four 4-bit pallet registers provided inside the unit, it permits choice of R, G, B, and I combinations, and it enables to make choice of any desired two or four colors out of sixteen available colors. However, in the 320 × 200, 16-color mode, choice of colors permitted to four kinds of colors output of sixteen.
- Only the conventional mode is applicable for the MZ-700 mode without using pallet.
- Pallet is not applicable for the border color.

<Configuration>



<Pallet register write> (F0_H)



1) S₀ ~ S₂: Register section

S ₂	S ₁	S ₀	Register No.
0	0	0	PLT 0
0	0	1	PLT 1
0	1	0	PLT 2
0	1	1	PLT 3
1	0	0	SW ₀ , SW ₁

2) B_i, R_i, G_i, I_i: Pallet write data

3) SW₀, SW₁:

With these switches, it is possible to make combination of Planes III and IV data in the 320 × 200, 16-color mode. Switches are used to assign pallets to four groups of colors.

(Plane III data) = SW₀, (Plane IV data) = SW₁

Only for the color information, the color information set by the pallet register are available as B, R, G, and I outputs. For color information other than that, data in Plane I through Plane IV are sent out as the B, R, G, and I outputs.

(See example next.)

<Pallet output and display mode>

- Shown next is the relation of the display mode, color plane data vs R, G, G, I outputs.

Display mode			Display color	Pallet output select		Pallet enable SW ₀ , SW ₁	Output select		Output	Output	Output	Output	
				A	B		A	B					
320 × 200	4 color	Frame A	4 colors out of 16 colors	Plane I data	Plane II data	×	0	0	B ₀	R ₀	G ₀	I ₀	
							1	0	B ₁	R ₁	G ₁	I ₁	
		0	1	B ₂	R ₂		G ₂	I ₂					
		1	1	B ₃	R ₃		G ₃	I ₃					
	Frame B	4 colors out of 16 colors	Plane III data	Plane IV data	×	0	0	B ₀	R ₀	G ₀	I ₀		
						1	0	B ₁	R ₁	G ₁	I ₁		
						0	1	B ₂	R ₂	G ₂	I ₂		
						1	1	B ₃	R ₃	G ₃	I ₃		
	16 colors	—	(Ex.) 16 colors out of 16 colors	Plane I data	Plane II data	SW ₀ = $\left\{ \begin{array}{l} \text{Plane III} \\ \text{data} \end{array} \right\}$	0	0	B ₀	R ₀	G ₀	I ₀	
							1	0	B ₁	R ₁	G ₁	I ₁	
							0	1	B ₂	R ₂	G ₂	I ₂	
							1	1	B ₃	R ₃	G ₃	I ₃	
SW ₁ = $\left\{ \begin{array}{l} \text{Plane IV} \\ \text{data} \end{array} \right\}$						×	×	I	II	III	IV		
						×	×	I	II	III	IV		
						×	×	I	II	III	IV		
						×	×	I	II	III	IV		
640 × 200	2 colors	Frame A	2 colors out of 16 colors	Plane I data	×	×	0	×	B ₀	R ₀	G ₀	I ₀	
							1	×	B ₁	R ₁	G ₁	I ₁	
		Frame B	2 colors out of 16 colors	Plane III data	×		×	0	×	B ₀	R ₀	G ₀	I ₀
								1	×	B ₁	R ₁	G ₁	I ₁
	4 colors	—	4 colors out of 16 colors	Plane I data	Plane III data	×		0	0	B ₀	R ₀	G ₀	I ₀
								1	0	B ₁	R ₁	G ₁	I ₁
							0	1	B ₂	R ₂	G ₂	I ₂	
							1	1	B ₃	R ₃	G ₃	I ₃	

(Ex.)

An example of the pallet in use in the 320 × 200, 16-color mode

- Assume that the pallet register has been set to the following.

PLT0 = Black
PLT1 = Cyan
PLT2 = Red
PLT3 = Magenta

- When SW₀ is set to "0" and SW₁ to "0", the pallet is applied to four colors in group 1 (III = 0, IV = 0) and it results in the color as shown in ① of the table right (yellow to cyan).
- When SW₀ is set to "0" and SW₁ to "1", four colors of group 3 (III = 0, IV = 1) becomes the display color set by the pallet.
- Therefore, any color can be chosen out of 16 colors against four colors of color group selected by SW1 and SW2.
- For group other than selected by SW₀ and SW₁, the color that I ~ IV outputted on B, R, G, I is displayed.

Group	Plane data				Display color of I ~ IV → RGBI	SW ₀ = 0 SW ₁ = 0	SW ₀ = 0 SW ₁ = 1
	I	II	III	IV			
Group 1	0	0	0	0	Black	PLT0 = Black	Black
	1	0	0	0	Blue	PLT1 = Cyan	
	0	1	0	0	Red	PLT2 = Red	
	1	1	0	0	Magenta	PLT3 = Magenta	
Group 2	0	0	1	0	Green	←	←
	1	0	1	0	Cyan	←	←
	0	1	1	0	Yellow	←	←
	1	1	1	0	White	←	←
Group 3	0	0	0	1	Gray	←	PLT0 = Gray
	1	0	0	1	Light blue	←	PLT1 = Light blue
	0	1	0	1	Light red	←	PLT2 = Light red
	1	1	0	1	Light magenta	←	PLT3 = Light magenta
Group 4	0	0	1	1	Light green	←	←
	1	0	1	1	Light cyan	←	←
	0	1	1	1	Light yellow	←	←
	1	1	1	1	Light white	←	←

Border color

- As the CRTC has a 4-bit border color register, it permit to use any border color out of 16 colors.
- Border register (OUT 06CF_H)

	MSB				LSB			
BCOL	x	x	x	x	I	G	R	B

- B, R, G, and I becomes "0" (black) when reset.

4-9 CRTC register map

- VRAM control
- Data display on the video screen

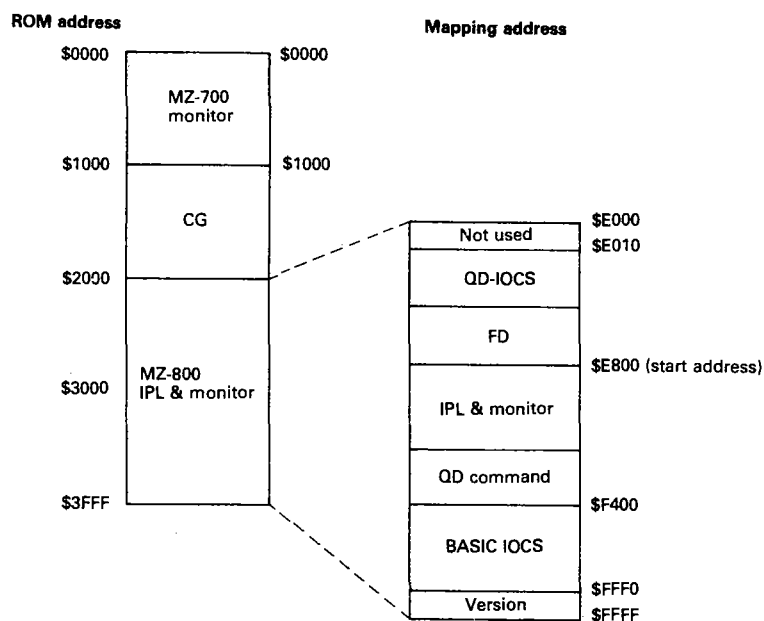
Control I/O address map

I/O address		IN/OUT	
H (B)	L (C, ☆)		
—	CC	O	Write format register (WF)
—	CD	O	Read format register (RF)
—	CE	O	Display mode register (DMD)
—	CE	I	Status read
01	CF	O	Scroll offset register L (SOF1), 8 bits
02	CF	O	Scroll offset register R (SOF2), 2 bits
03	CF	O	Scroll width register (SW), 7 bits
04	CF	O	Scroll start address register (SSA), 7 bits
05	CF	O	Scroll end address register (SEA), 7 bits
06	CF	O	Border color register (BCOL), 4 bits
07	CF	O	Superimpose bit (D7) (CKSW), 1 bit
	FD	O	Pallet register

Written by indirect
OUT command.
B register ← 0~7
OUT(C),A

4-10 ROM configuration

The MZ-700 monitor, character generator (CG), MZ-800 monitor, and IPL are implemented on a single chip of 16k × 8-bit ROM.



5 8255 Programmable Peripheral Interface

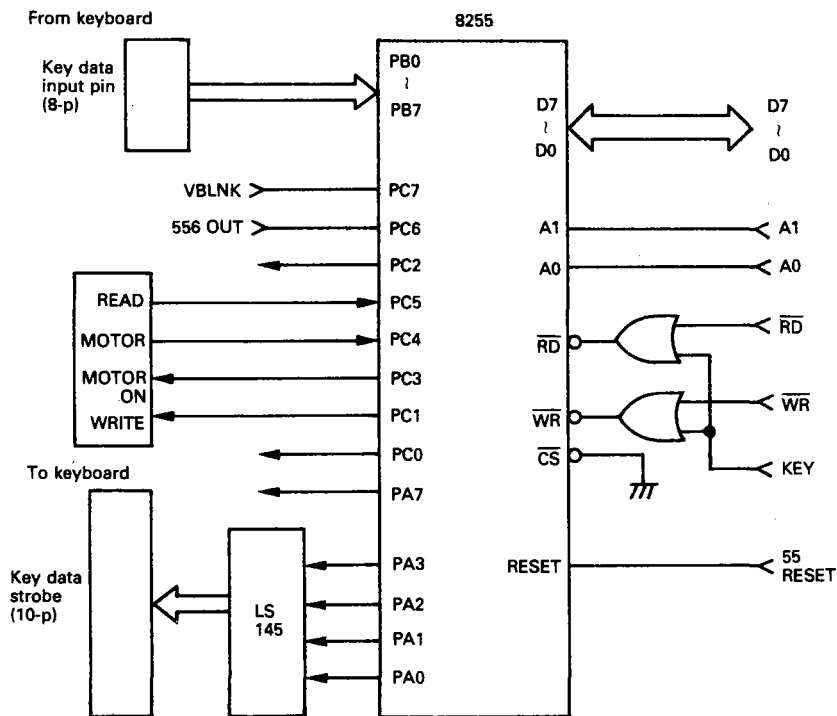
The 8255 has three pairs of 8-bit I/O ports, each one can be assigned to input or output port by means of programming. A different mapping is established de-

pending on the mode. In the MZ-700 mode, it is on memory space, and in the MZ-800 mode, it is on I/O space.

Port name (address)	Pin No.	I/O	Active state	Function
PA (700 \$E000) (800 \$D0)	PA ₀ PA ₁ PA ₂ PA ₃ PA ₄ PA ₅ PA ₇	O	H H H H L L L	Keyboard scan strobe Joystick-1 strobe Joystick-2 strobe CRT cursor blink timer reset
PB (700 \$E001) (800 \$D1)	PB ₀ PB ₁ PB ₂ PB ₃ PB ₄ PB ₅ PB ₆ PB ₇	I	L L L L L L L	Keyboard scan input
PC (NOTE-1) (700 \$E002) (800 \$D2)	PC ₀ PC ₁ PC ₂ PC ₃ PC ₄ PC ₅ PC ₆ PC ₇	O O O O I I I	L L L L H L L	Prohibits sound output of the 8253 Cassette write data Disables timer interrupt Rotates the cassette motor Checks the cassette motor Cassette read data CRT cursor blink timer input Vertical blink signal
(700 \$E003) (800 \$D3)	—	—	—	Control por

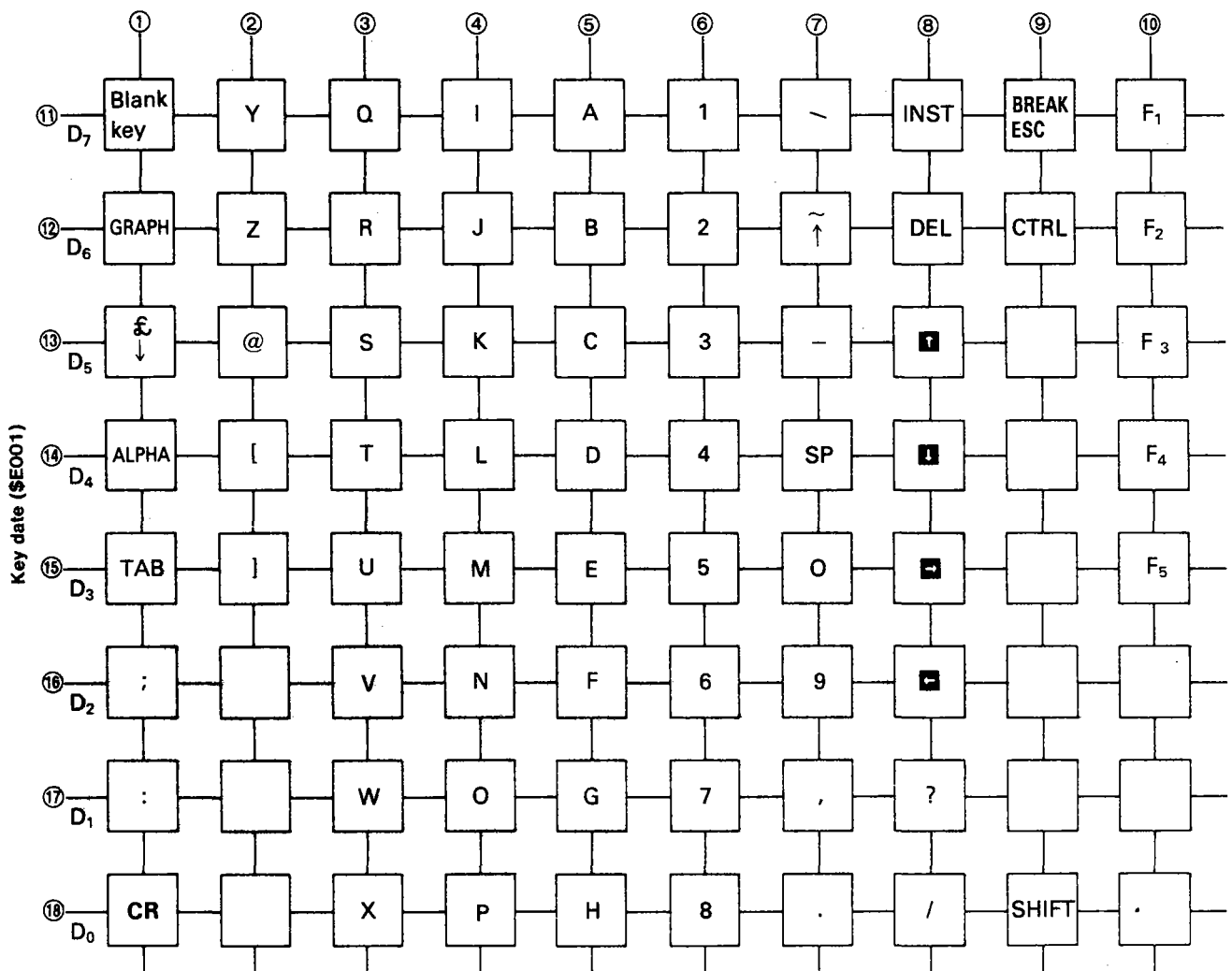
NOTE-1: Output data dependent on the bit set mode.

NOTE-2: Motor is controlled on and off by the rising edge of the signal.



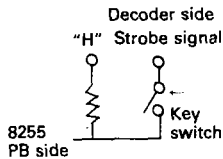
a) Key scan
Ports PA₀~PA₃ of the 8255 are connected via the

LS145 decoder, and PB₀~PB₇ are connected to the key matrix directly.



Key strobe is issued through PA₀~PA₃ to scan the key. As it is supplied to the decoder, it makes one of outputs, 0~9, set low. It is then added to the key matrix to scan the line of the key depressed (vertical key matrix scan). The line is in the low state, if it is in depression (horizontal key matrix scan).

NOTE: In the ready for command state, PA₀~PA₃ are normally repeats to be low state and the decoder outputs repeats to be high state. But, since the decoder is of an open collector type, it would not permit to check high and low state.



Example

8255 output PA₃ PA₂ PA₁ PA₀
L H L L

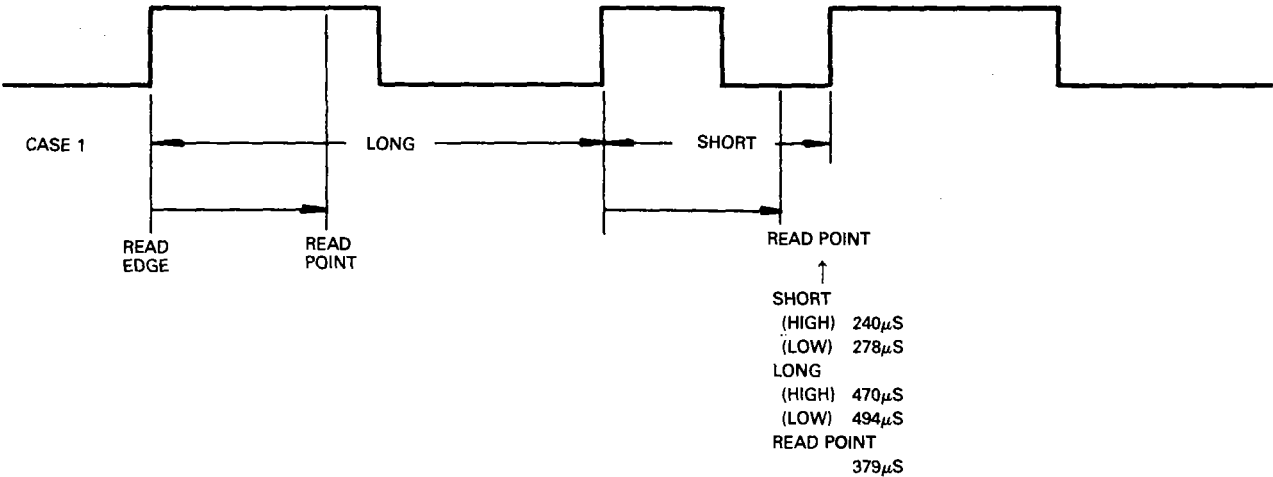
Only the LS145 decoder output 4 (#5 pin) is in low state.

Because the connector (5) is in low state, key scan is permitted only for keys, A through H.

State of the 8255 input port B.
PB₇ PB₆ PB₅ PB₄ PB₃ PB₂ PB₁ PB₀
L H H H L H H L
Above state shows that keys, A, E, or H, is in depression.

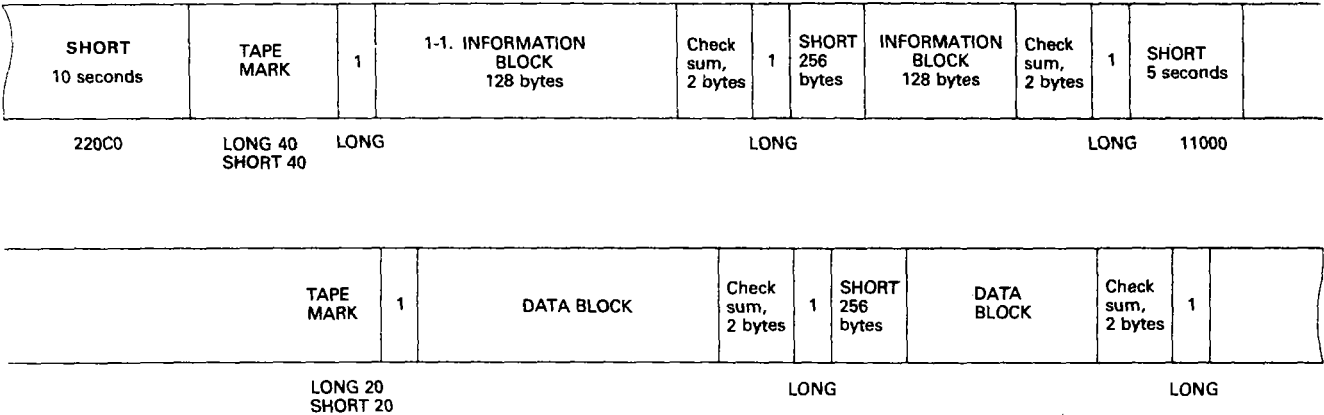
b) Cassette control
The 8255 issues the cassette write data from PC1 and

read signal through PC5. The type of data (input, output) and its format are as follows:



LONG represents the bit value "1" and SHORT the bit value "0". Data will be read at 368 microseconds after

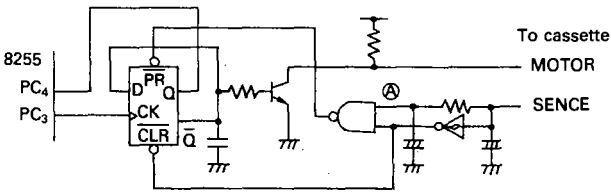
the signal rising edge. Data are recorded in repetition of LONG and SHORT, and the same data are written twice.



See next for the contents of the information block.

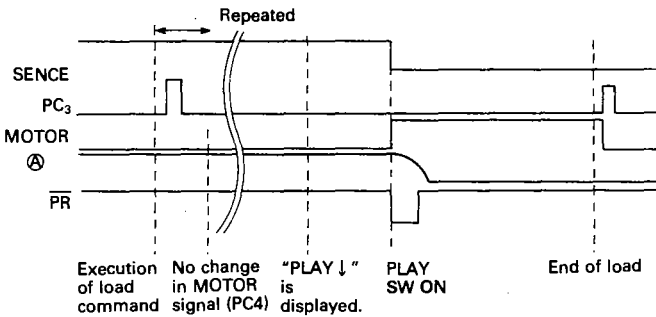
Name	Byte count	Function	Note
ATRB	1	Attribute	
NAME	17	File name (16 characters maximum)	CR (0D) affixed
SIZE	2	File byte size	In order of low to high order
DTADR	2	Loading address	
EXADR	2	Execution address	
COMNT	104	Comment	Not used

Rotation of the cassette (dedicated) is controlled by the 8255 and its peripheral circuits.

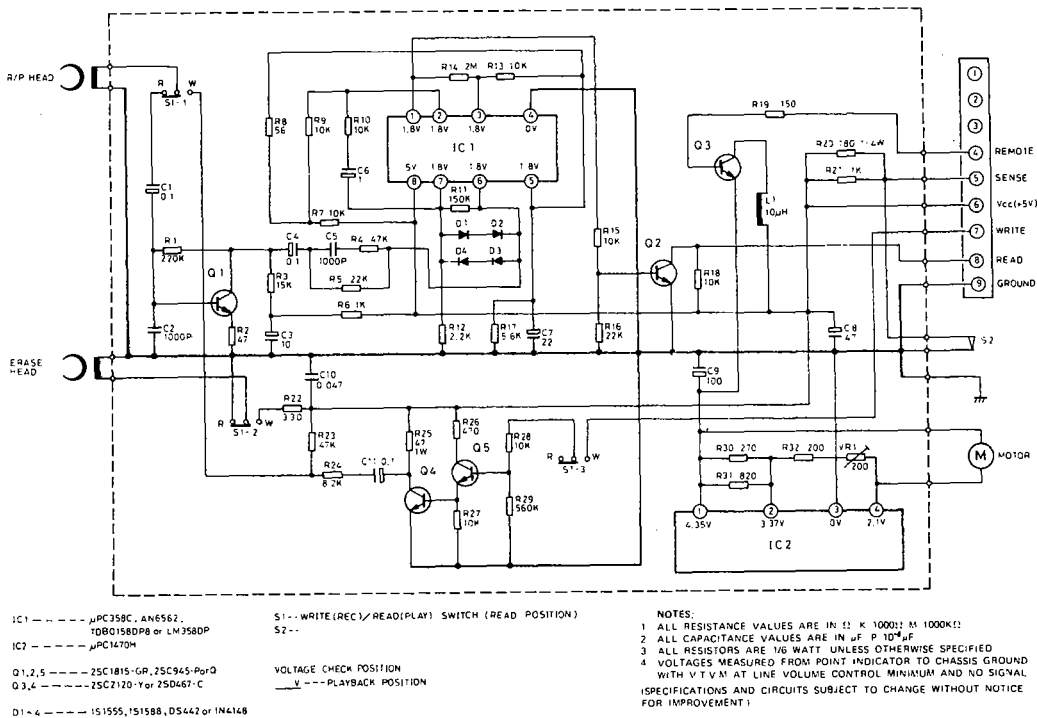


If switch has not been ON on the cassette recorder side, SENSE signal is in high state. When a switch (REW, FF, etc.) is pushed, it makes the signal turned low. It presets the D-FF and the motor starts to rotate with MOTOR in high state. With lock given to the D-FF through PC3, it permits on/off control of the motor. If a switch is pushed on the cassette recorder side, it permits examination of the motor operating state by means of L and PC4.

For use of other than MZ-800 cassette tape recorder type, it needs to short SENSE to GND, READ to EXREAD, and WRITE to EXWRITE of the connector T-5. Use of the cassette recorder of other kind may sometimes not permit proper loading and saving operation. In such an event, adjust the volume and tone controls to find the optimum positions. To meet the opposite polarity of cassette tape recorder, there is a dip switch provided. Changing the switch position makes TPSW signal state changed so as to invert the signal waveform.



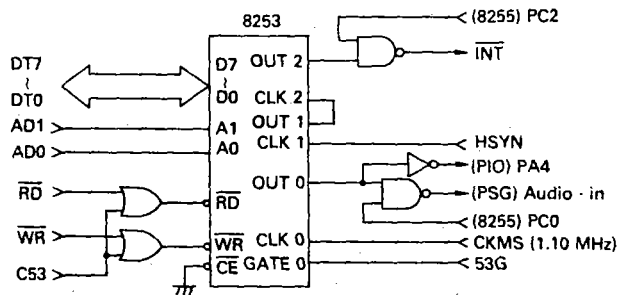
Casset Recorder Circuit



6 8253 Programmable Interval Timer

The 8253 makes sound generated with the counter #0 and internal timer is operated with the counters #1 and #2.

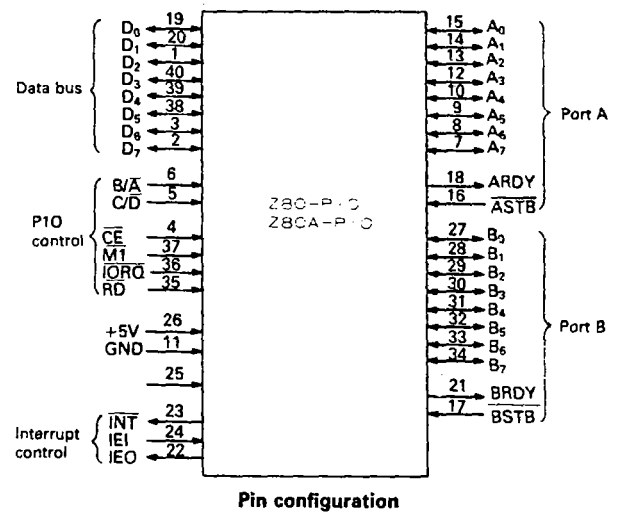
- Counter mode
 - #0 Square waveform generator MODE3
 - #1 Rate generator MODE2
 - #2 Interrupt on terminal counter



- The counter #0 counts input pulse of 1.1 MHz, divided by the predetermined rate (musical score data) to generate sound. It is connected with the mixing audio amplifier through AUDIO-IN of the sound IC (76489AN). This counter output is gated by PC of the 8255 port C, and the counter gate is controlled by D01 of \$E008. The counter #0 output is also used for interrupt control INT0 and connected to A4 of the Z-80A PIO port A.
- The counter #1 counts pulse of 15.6 kHz and generated a pulse on OUT1 at every second. The counter #2 counts pulses and makes OUT2 turned high. OUT2 outputs becomes $\overline{\text{INT}}$ via the gate and is connected to INT of the CPU.

7 Printer interface

The Z-80A PIO is used for the printer interface. It has a pair of 8-bit I/O ports.



Pin name	Pin No.	I/O	Signal name	Description															
D ₀ ~D ₇	19,20,1 40,39,38 3,2	I/O	Z80-CPU Data Bus	Bidirectional, 3-state, Z-80 CPU bus. Data and command transfer between the Z-80 CPU and the PIO is carried out through this data bus. D ₀ is the least significant digit.															
B/ $\overline{A}$	6	I	Port B or A Select	Port select signal. Depending on the state of this signal, the port is specified through which data or command is transferred between the Z-80 CPU and the PIO. } H : Port B } L : Port A															
C/ $\overline{D}$	5	I	Control or Data Select	Control/data select signal. Depending on the state of this signal, control port or data port is selected for the port assigned with B/ $\overline{A}$. <table><tr><th>B/$\overline{A}$</th><th>C/$\overline{D}$</th><th>Selected port</th></tr><tr><td>L</td><td>L</td><td>Port A data</td></tr><tr><td>L</td><td>H</td><td>Port A control</td></tr><tr><td>H</td><td>L</td><td>Port B data</td></tr><tr><td>H</td><td>H</td><td>Port B control</td></tr></table>	B/ $\overline{A}$	C/ $\overline{D}$	Selected port	L	L	Port A data	L	H	Port A control	H	L	Port B data	H	H	Port B control
B/ $\overline{A}$	C/ $\overline{D}$	Selected port																	
L	L	Port A data																	
L	H	Port A control																	
H	L	Port B data																	
H	H	Port B control																	
$\overline{CE}$	4	I	Chip Enable	Chip enable signal. A low on this line enables the PIO. Normally connected with the I/O address decoder output.															
ϕ	25	I	System Clock	System clock CPU clock ϕ is usually used.															
$\overline{M1}$	37	I	Machine Cycle One	Connection with CPU $\overline{M1}$ signal (low active). The PIO attains synchronization with the CPU interrupt control logic by $\overline{M1}$. The PIO will be reset when $\overline{M1}$ is set low at least for a period of two clock cycles after turning $\overline{IORQ}$ and $\overline{RD}$ high state.															
$\overline{IORQ}$	36	I	Input Output Request	Connection with CPU $\overline{IORQ}$ signal (low active). This signal perform data transfer between the CPU and the PIO in connection with B/ $\overline{A}$, C/ $\overline{D}$, $\overline{CE}$, and $\overline{RD}$. If $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are low, the data on the port selected by B/ $\overline{A}$ are transferred to the CPU. If $\overline{CE}$, $\overline{IORQ}$ are low, data or command is written through the port selected by B/ $\overline{A}$.															
$\overline{RD}$	35	I	Read	Connection with CPU $\overline{RD}$ signal (low active). This signal controls the direction of data transfer between the CPU and the PIO in connection with B/ $\overline{A}$, C/ $\overline{D}$, $\overline{CE}$, and $\overline{IORQ}$.															
IEI	24	I	Interrupt Enable in	Interrupt daisy chain signal. The PIO will respond to the INTA cycle of the CPU only when this signal is high.															
IEO	22	O	Interrupt Enable Out	Interrupt daisy chain signal. This signal is high only when IEI is not high with the PIO having an interrupt request. It goes low when IEI is low or PIO is having an interrupt request.															

Pin name	Pin No.	I/O	Signal name	Description
INT	23	O	Interrupt Request	Connection with CPU INT signal. A low on this line causes the PIO to place an interrupt request to the CPU. Because it is of an open drain type, it is possible to make INT of several peripheral LSI wired OR using the pullup resistance.
A ₀ ~A ₇	15~12 10~ 7	I/O	Port A Bus	Port A data bus. Data transfer is carried out with the PIO and peripheral device via this bus. A0 is the least significant digit.
ASTB	16	I	Port A Strobe	Port A strobe. Significance of this signal depends on the Port A operational mode. 1) Byte output mode : It indicates that the peripheral device has received data from the PIO at a rising edge of this strobe. 2) Byte input mode : Peripheral device loads data in the PIO port A input data register at a rising edge of this strobe. 3) Bidirectional mode : The contents of the port A output data register are outputted on A0~A7 when the strobe is in low state. 4) Bit mode : Not used.
ARDY	18	O	Register A Ready	Register A ready. Significance of this signal depends on the state of the port A operational mode. 1) Byte output mode : Data are loaded in the port A data output register when this signal goes high, makes A0~A7 stable, and it indicates that data can be transferred to a peripheral device. 2) Byte input mode : A high on this line indicates that the port A data input register is not occupied so as to be ready for receiving of a next data into the data register. 3) Bidirectional mode : This signal is used to indicate that data has been ready in the port A output data register. Data will not be issued on A0~A7 in this mode, unless ASTB turns low. 4) Bit mode : Not used.
B ₀ ~B ₇	27~34	I/O	Port B Bus	Port B data bus. Function of this bus is identical to A0~A7. But, it permits to drive a Darlington transistor as the bus can supply 1.5 V, 1.5 mA. B0 is the least significant digit.
BSTB	17	I	Port B Strobe	Port B strobe. Function of this signal is identical to ASTB, except for the following: This signal is used to load data from a peripheral device into the port A input data register, when the port A is in the bidirectional mode.
BRDY	21	O	Register B Ready	Register B ready. Function of this signal is identical to ARDY, except for the following: This signal indicates that the port A input data register is unoccupied and is ready for receiving of a next data, when the port A is in the bidirectional mode.

For the MZ-800 the PIO located on the I/O space, and address of ports performs the following:

\$FC Port A control

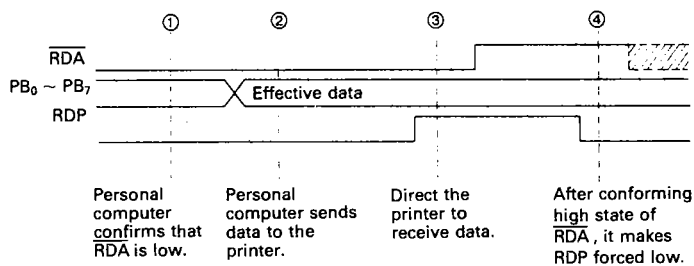
\$FD Port B control

\$FE Port A data

\$FF Port B data

	Pin name	I/O	Active	Signal name	Function
Port A (\$FE)	PA ₀	IN	L	$\overline{RDA}$	A low on this line indicates that the printer data is ready to receive. A low on this line informs the personal a paper depletion during status check.
	PA ₁	IN	L	$\overline{STA}$	
	PA ₂	IN	—	GND	
	PA ₃	IN	—	GND	
	PA ₄	IN	H		An 8253 output used for interrupt. Horizontal blanking signal used for interrupt. Used for printer initialization. Indicates the printer to receive data.
	PA ₅	IN	H		
	PA ₆	OUT	H	IRT	
	PA ₇	OUT	H	RDP	
Port B (\$FF)	PB ₀	OUT	—	RD ₀	Printer data or control code to the printer.
	PB ₁	OUT	—	RD ₁	
	PB ₂	OUT	—	RD ₂	
	PB ₃	OUT	—	RD ₃	
	PB ₄	OUT	—	RD ₄	
	PB ₅	OUT	—	RD ₅	
	PB ₆	OUT	—	RD ₆	
	PB ₇	OUT	—	RD ₇	

Interfacing timing



After the personal computer confirms that the printer is ready to receive data at (①), the data is then sent to \$FF port (PIO port B) at step (②).

As reception of data is directed to the printer at step (③), it makes RDP forced low at step (④) upon confirming that the printer received it ($\overline{RDA}=H$). After this, it awaits until $\overline{RDA}$ goes from high to low before transfer of a next data. But, it is possible to transfer successive data by interrupting the CPU at a falling edge of $\overline{RDA}$, since $\overline{RDA}$ is inputted to the $\overline{RSTB}$ input of the PIO, when in the port B mode 0.

It is also possible to interrupt the CPU referring to Port A inputs. Though discussed above is the printer interface method for the MZ compatible printer types, there is the Centronics compatible method for parallel interfacing of the printer. Since this method is basically the same as the MZ mode, except that signal polarity is opposite.

MZ specification		Centronics specification	
Signal name	Active	Signal name	Active
$\overline{RDA}$ (NOTE)	"L"	BUSY	"H"
$\overline{RDP}$	"H"	$\overline{STB}$	"L"
IRT	"H"	INPUT PRIME	"L"

NOTE: Though $\overline{RDA}$ is active low, it may be handled the same as high state of BUSY when considered in term of signal significance.

As shown in the figure above, it could be known that $\overline{RDP}$ and IRT should be inverted in order to make connection with the Centronics compatible printer. It can be attained by changing PRSW to high using the dip switch.

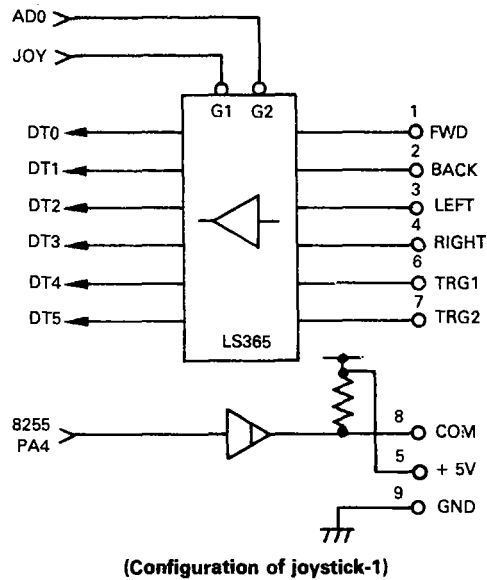
NOTE: When the MZ-800 dedicated printer is used, there may be such a case that proper operation is not attained due to different printing characters and control codes. It must be also noted that all MZ-800 characters can not be printed. Besides, connection with a Centronics compatible printer may not be permitted hardware-wise, sometimes.

8 Programmable sound generator

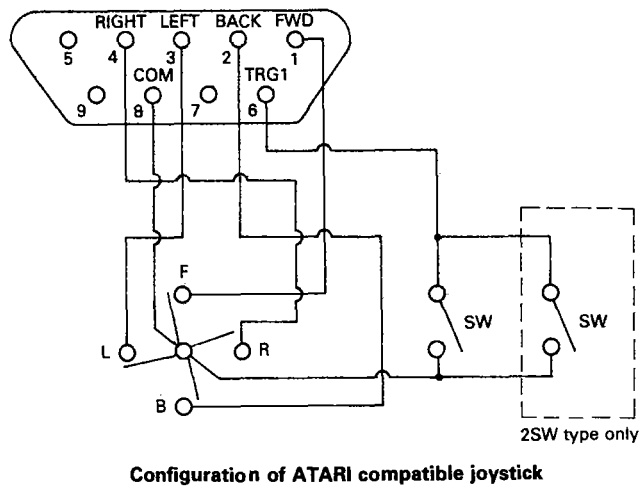
The SN76489N is used for the programmable sound generator which is controlled by the I/O port \$F2. It is write only. In order to permit smooth sound generation, timer interrupt is applied using the 8253. For the interrupt timer, the count 0 of the 8253 is used. The counter 0 is used for creation of sound steps in the MZ-700 mode, but, it is used for the timer interrupt source of the PSG in the MZ-800 mode. Interrupt is controlled by PA5 of the PIO. It is, however, possible to mask the counter 0 output by PC0 of the 8255, in order to prevent sound generation during interrupt.

9 Joystick

It permits connection of two Atari compatible joysticks.



Low active strobes are issued through PA4 (JOY1) of the 8255 and PA5 (JOY2) interrogate switch activation through inputs to \$F0 (JOY1) and \$F1 (JOY2).



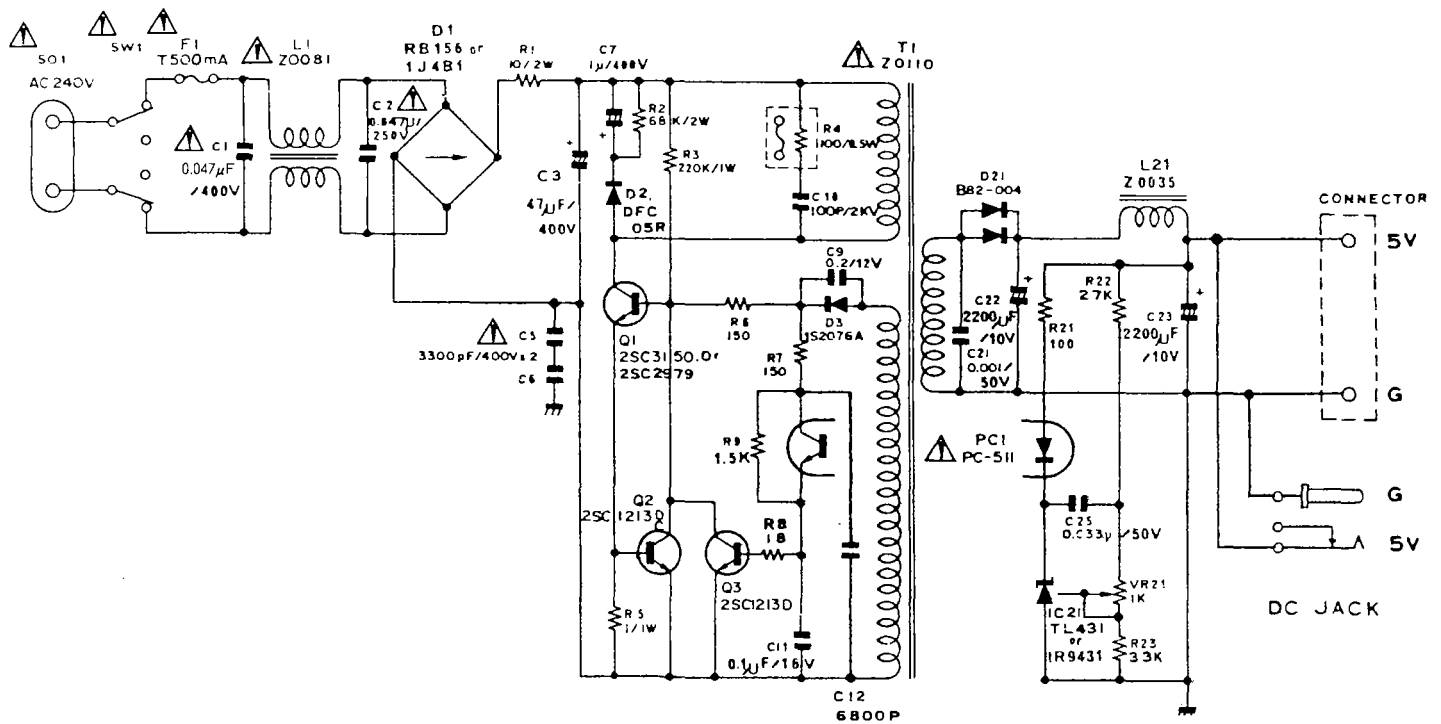
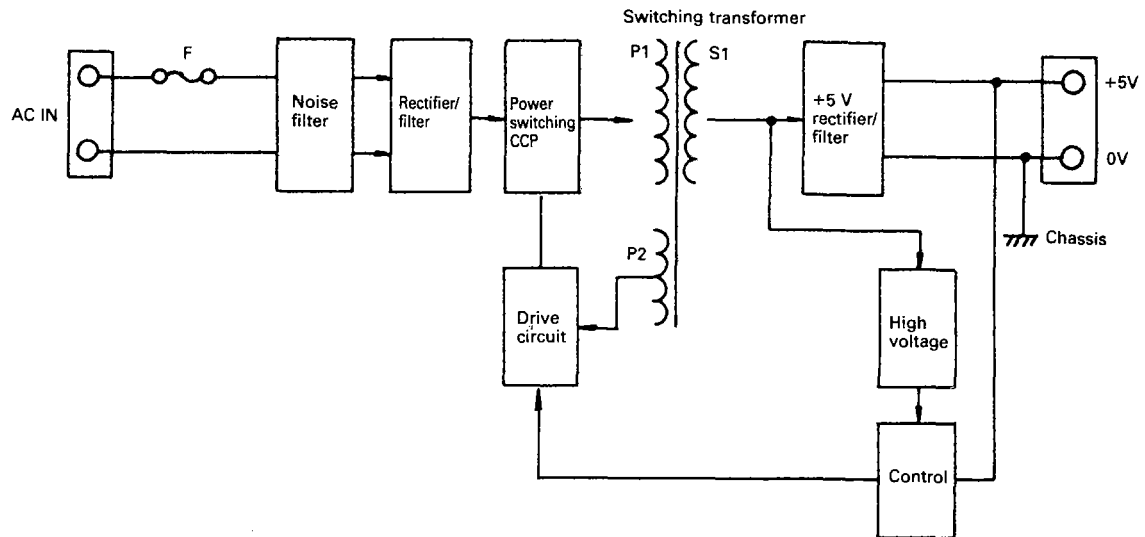
System switches are assigned as follows:

SW No.	Function	Setup method
1	MZ-700/MZ-800 selection	ON: MZ-700 mode OFF: MZ-800 mode
2 3	MZ/Centronics printer selection	MZ printer with SW2 and SW3 at ON Centronics printer with SW2 and SW3 at OFF
4	External cassette recorder polarity selection	Changed so as to enable read on the external cassette recorder.

* Switch setups at the factory
SW1 OFF (MZ-800 mode)
SW2 ON
SW3 ON } (MZ dedicated printer)
SW4 ON

10 Power supply

10-1 Block diagram



10-2 Operational description

The block diagram of the power supply unit is shown above. It adopts the self-excitation ON/OFF control method. First, the source supply is rectified through the noise filter and converted into direct current. As the dc current is applied to the switching transistor, it causes the transistor to start repeating ON and OFF. After the dc voltage is converted into high frequency pulse, it is added to the primary side of the main transformer which causes to induce voltage on the secondary side. This high frequency pulse is then rectified and filtered to obtain the dc voltage of +5 volts. For control of output voltage, the output voltage is compared with the reference voltage and its error is detected in the control section. While the switching transistor is in the OFF cycle, it makes the photo coupler PC1 active by the detecting signal of the control section for given period. By adjusting control current of the drive circuit, it makes the output stabilized.

10-3 Maintenance

Cleaning

Dust deposit inside the power supply unit may become the cause for overheating as it prevents heat dispersion, which results in damage to components. Stains on the fuse contact and connector contact may lead to contact failure. So, it has to be cleaned using soft cloth dampened with alcohol or dry soft cloth.

10-4 Problem determination and sequence

Follow the next procedure to find the cause of trouble.

- (1) Avoid removing the board to check. But visually observe the board to check for open circuit line, burnt resistor, fuse, and semiconductor chips in the first place.
- (2) If a defective item were found, it has to be replaced with the new one. But, care must also be taken as there are possible defects in multiple numbers of components.

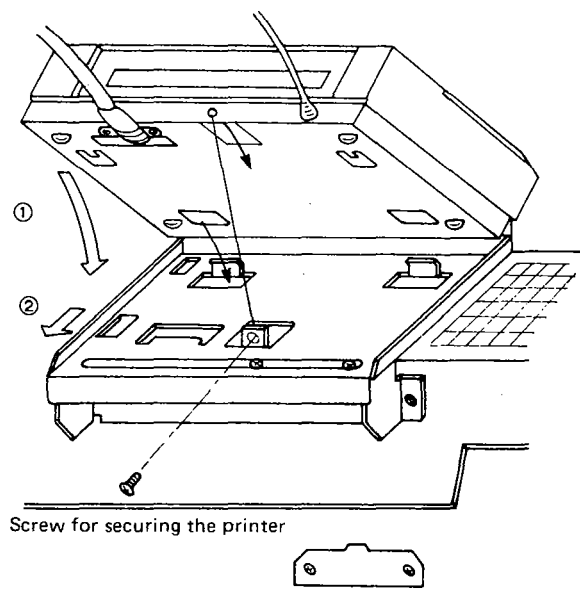
11 MZ-1P16

11-1 Installation

1. Fix the printer unit on the table as shown in the figure.

(Place the printer in the arrow direction ①, lightly move in the arrow direction B, then secure it with screws.)

* It is also possible to use the printer free on the table without securing.

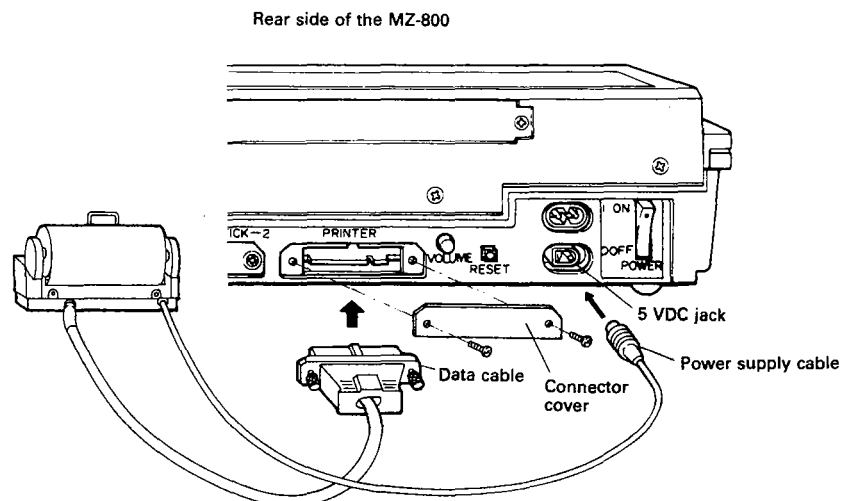


Connection procedure after the installation

- ① Make sure that power is off to the MZ-800 and its peripheral units.
- ② Remove the printer connector cover on the back side of the MZ-800.

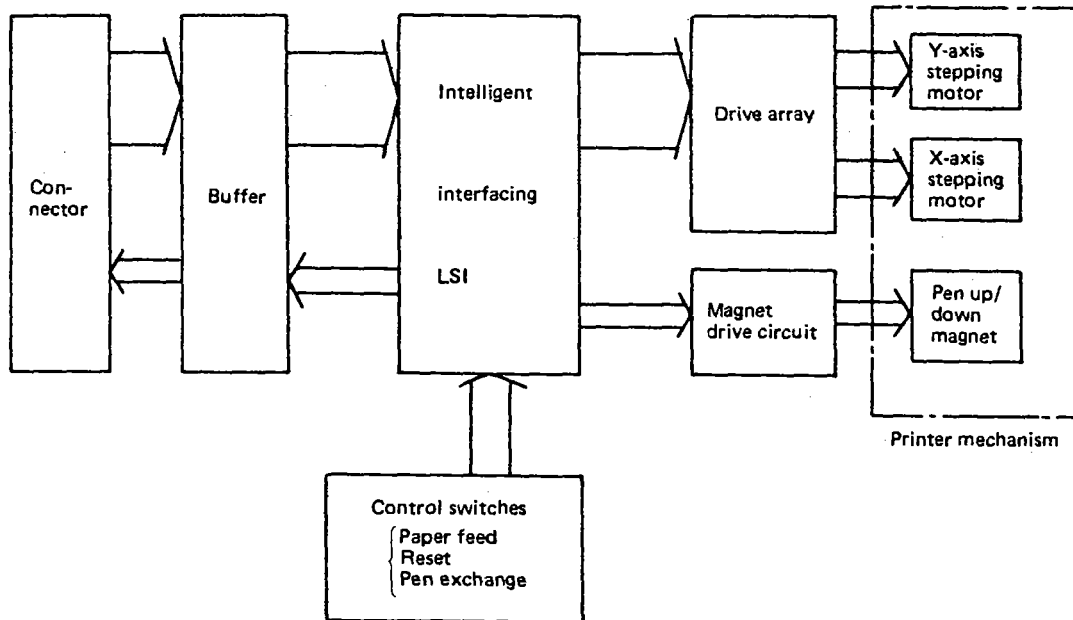
- ③ Connect the data cable and the power supply cable of the unit with the printer connector and the 5 VDC jack of the MZ-800.

- ④ For connection of the printer connector, use the screws that had been at both ends of connector.



* When this unit is in use, set the MZ-800 printer dip switch to the MZ side.

11-2 Block diagram



11-3 At power on

At power on, more than 5V of pen up current is applied for a period of 10ms, plus 5 and minus 0ms, to move the carriage 556 steps backward on the X-axis in order to initialize the colour position. As the carriage is held at the left margin after disengagement of the motor, it is then moved 30 steps forward on the X-axis, then stepped back 30 steps again to check if the colour position detector has been made. If not, it continues to move the carriage 30 steps forward on the X-axis, then return 30 steps to ensure the made condition.

11-4 Colour change operation

To change colour, the slider makes three reciprocating movements of 6mm (30 steps) at the left end of the X-axis to move the pen position one step. When the desired pen position is attained, it then returns to the home position.

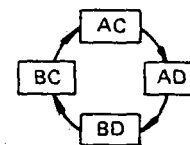
Since the pen rotor makes a unidirectional rotation at the left end of the X-axis, and is locked within printable range, care must be exerted not to touch the rotor and the slider.

11-5 Pen exchange operation

A pen needs to be exchanged with a fresh one when it runs out of ink. In such an event, the pen is moved 485 steps forward on the X-axis from the home position with the used pen located on the top of the rotor, then take out the used pen, by pressing the pen release lever and exchange it with a fresh one.

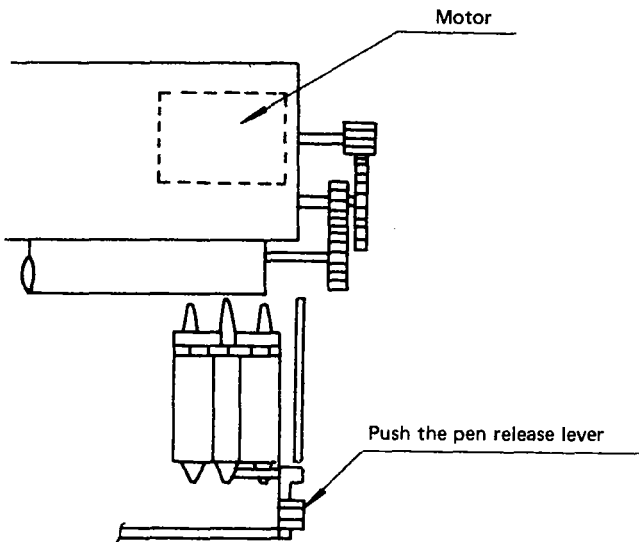
Motor phase and rotating direction

The arrow head indicates the forward direction for both the X-axis and Y-axis.



11-6 Pen exchange method

To remove pen, press the pen exchange button, when the slider is at the right handside, push the pen release lever.



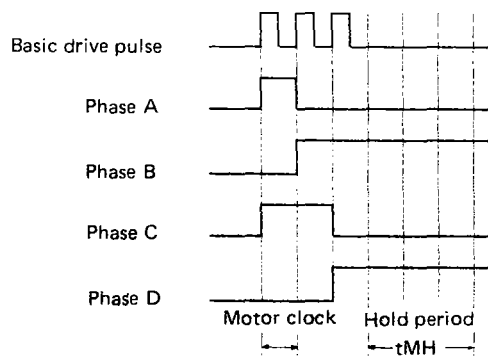
To install the pen, push the tip of the pen through the ring of the return spring in the first place, then push into the holder. Upon completion, ensure that the tip of the pen is engaged with the hole of the pen return spring.

If colour change is done when the pen is disengaged from the hole, it may cause improper rotation of the rotary holder as the slider makes contact with the pen. Do not try to rotate the rotary holder by hand when installing the pen during replacement of the pens.

11-7 Stepping motor driving signal

- The X-axis stepping motor and the Y-axis stepping motor are driven by the two-phase magnet.

Stepping motor driving signal



It is more effective to save power to shut off current while the X and Y axis motors are at a halt. But, there may be a possible malfunction because of unsuppressed vibration, if the current is turned off with a normal pulse width. In order to prevent this, current is applied excessively for more than the given hold time ($t_{MH} = 1\text{ms}$ or more).

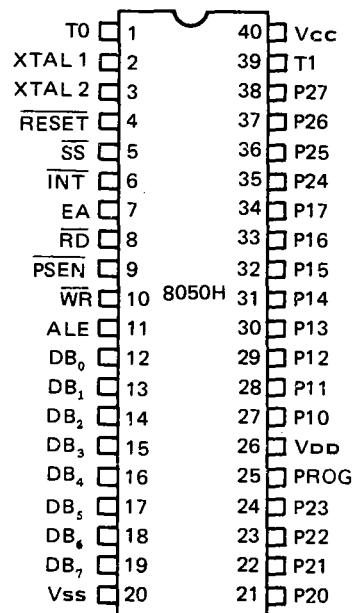
11-8 Colour position detector

The colour position detector consists of a reed switch and a permanent magnet and it may cause malfunction owing to external vibration, and magnetic influence. Especially, when deposit of alien matter or paper fragments is between the left end of the carriage and the frame this may result in a failure of the colour detect performance.

11-9 Character set

Input of an undefined code up to \$20 is ignored. Other undefined codes are represented in hexadecimal notation using the pen in a next colour position.

Pin configuration (top view)



Pin Configuration

11-10 Colour plotter printer control LSI

Pin assignment

Symbol	Name	In/out	Function
V _{SS}	Ground		Connected to 0V.
V _{CC}	Main power		Connected to 5V.
V _{DD}	Power		Connected to 5V.
PROG	Program	Out	Not used.
P ₁₀ ~ P ₁₇	Port 1		Used as printer control signals.
P ₂₀ ~ P ₂₇	Port 2		Used for data input port from CPU.
D ₀ ~ D ₇	Data bus		Used for stepper motor control signals.
T ₀	Test pin 0	In	Input from pen change switch.
T ₁	Test pin 1	In	Input from paper feed switch.
$\overline{\text{INT}}$	Interrupt input	In	Data transfer strobe MZ-700 → MZ1P01.
$\overline{\text{RD}}$	Read signal	In	Not used.
$\overline{\text{WR}}$	Write signal	Out	Not used.
$\overline{\text{RESET}}$	Reset	In	Used to initialize the processor.
ALE	Address latch enable	Out	Not used.
$\overline{\text{PSEN}}$	Program store enable	Out	Not used.
$\overline{\text{SS}}$	Single step	In	Not used.
EA	External access	In	Active when EA = 0V.
X ₁ , X ₂	Crystal inputs	In	Pins used to attach the crystal oscillator or RC network to generate internal clock. However, external clock signal may be inputted through these pins.

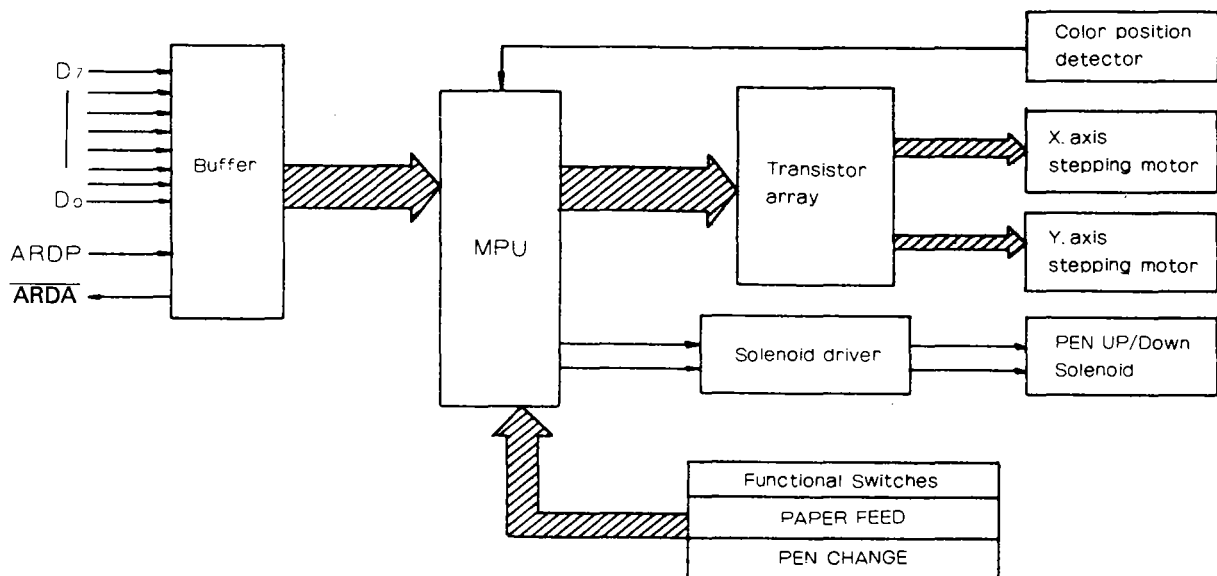
11-11 Interfacing with the MZ-800

Fig.1 shows the block diagram for connection with the printer. Fig.2 shows its circuit description. Fig.3 shows the timing chart.

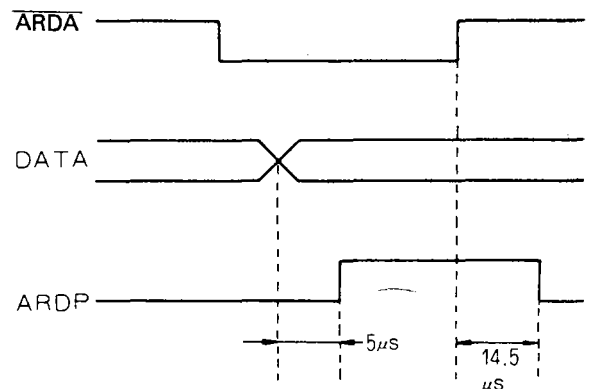
Table of character set

MSD LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P			}		q	n				
1		↓	0	1	A	Q					a					
2		↑		2	B	R				e	z	U				
3		→	#	3	C	S				`	w	m				
4		←	\$	4	D	T				~	s					
5		H	%	5	E	U					u					
6		⊙	&	6	F	V				t	i		→			
7			'	7	G	W				g		o		=		
8			(	8	H	X				h	ō	l				
9			)	9	I	Y					k	ā				
A			*	:	J	Z				b	f	ō				
B			+	;	K	[		°	^	x	v	ā				ℝ
C			,	<	L	\				d						↓
D			-	=	M	]				^	ü	y				
E			.	>	N	↑				p	β	{				
F			/	?	O	←				c	j		=			π

11-12 Block diagram



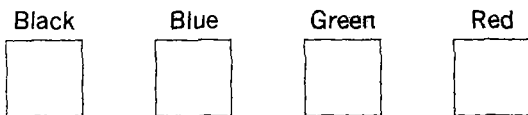
The CPU sends data to the printer after confirming that ARDA is in low state. Five micro seconds later, the strobe signal ARDP goes high. The CPU confirms that ARDA is in high state, ARDP is returned to a low state 14.5 micro seconds later.



11-13 Colour plotter printer control codes

Control codes used in the text mode

- Text code (\$01)
Places the printer in the text mode.
- Graphic code (\$02)
...Same as the BASIC PMODE statement. Places the printer in the graphic mode.
- Line up (\$03)
...Same as the BASIC PSKIP statement. Moves the paper one line in the reverse direction. The line counter is decremented by 1.
- Pen test (\$04)
...Same as the BASIC PTEST statement. Writes the following patterns to start ink following from the pens, then sets scale = 1 (40 chr/line), colour = 0.



- Reduction scale (\$09)+(\$09)+(\$09)
Reduces the scale from 1 to 0 (80 chr/line).
- Reduction cancel (\$09)+(\$09)+(\$0B)
Enlarges the scale from 0 to 1 (40 chr/line).
- Line counter set (\$09)+(\$09)+(ASCII)₂+(ASCII)₁+(ASCII)₀+\$0D
...Same as the BASIC PTEST statement. Specifies the number of lines per page as indicated by the 3 ASCII bytes code. The maximum number of lines per page is 255. Automatically set to 66 when the power is turned on or the system is reset.

- Line feed (\$0A)
...Same as the BASIC PTEST statement. Moves the paper one line in the forward direction. The line counter is incremented by 1.
- Magnify scale (\$0B)
Enlarges the scale from 2 to 1. (26 chr/line)
- Carriage return (\$0D)
Moves the carriage to the left side of the print area.
- Back space (\$0E)
Moves the carriage one column to the left. This code is ignored when the carriage is at the left side of the print area.
- Form feed (\$0F)
Moves the paper to the beginning of the next page and resets the line counter to 0.
- Next colour (\$1D)
Changes the pen to the next colour.

Character scale

- The character scale is automatically set to 1 (40 chr/line) when the power is turned on. Afterwards, it can be changed by the control codes and commands.
- In the graphics mode, the scale can be changed within the range 0 to 63.
- The scale is set to 1 when the mode is switched from graphics to text.

11-14 Graphic mode commands

Command type

In the graphics mode, the computer can control the printer with the following commands.

The words in parentheses are BASIC statements which have the same functions as the graphics mode commands.

Command name	Format	Function
LINE TYPE	Lp (p=0 to 15)	Specifies the type of line (solid or dotted) and the dot pitch. p=0 : solid line, p=1 to 15 : dotted line
ALL INITIALIZE	A	Places the printer in the text mode.
HOME (PHONE)	H	Lifts the pen and returns it to the origin (home position).
INITIALIZE (HSET)	I	Sets the current pen location as the origin (x=0, y=0).
DRAW (LINE)	Dx, y, ..., xn, yn (-999 ≤ x, y ≤ 999)	Draws lines from the current pen location to coordinates (x1, y1), then to coordinates (x2, y2), and so forth.
RELATIVE DRAW (RLINE)	JΔx, Δy, ..., Δxn, Δyn (-999 ≤ Δx, Δy ≤ 999)	Draws lines from the current pen location to relative coordinates (Δx1, Δy1), then to relative coordinates (Δx2, Δy2), and so forth.
MOVE (MOVE)	Mx, y (-999 ≤ x, y ≤ 999)	Lifts the pen and moves it to coordinates (x, y).
RELATIVE MOVE (RMOVE)	RΔx, Δy (-999 ≤ Δx, Δy ≤ 999)	Lifts the pen and moves it to coordinates (Δx, Δy).
COLOR CHANGE (PCOLOR)	Cn (n=0 to 3)	Changes the pen colour to n.
SCALE SET	Sn (n=0 to 63)	Specifies the character scale.
ALPHA ROTATE	Qn (n=0 to 3)	Specifies the direction in which characters are printed.
PRINT	Pc1c2c3 ... cn (n=∞)	Prints characters.
AXIS (AXIS)	Xp,q,r (p= 0 or 1) (q= -999 to 999) (r=1 to 255)	Draws an X axis when p=1 and a Y axis when p=0. q specifies the scale pitch and r specifies the number of scale marks to be drawn.

Command format

There are 5 types of command formats as shown below.

1. Command character only (without parameters)

A, H, I

2. Command character plus one parameter

L, C, S, Q

3. Command character plus pairs of parameters

D, J, M, R

", " is used to separate parameters, and a CR code is used to end the parameter list.

4. Command plus character string

P

The character string is terminated with a CR code.

5. Command plus three parameters

X

", " is used to separate parameters.

《MZ1P16》

11-15 Specification

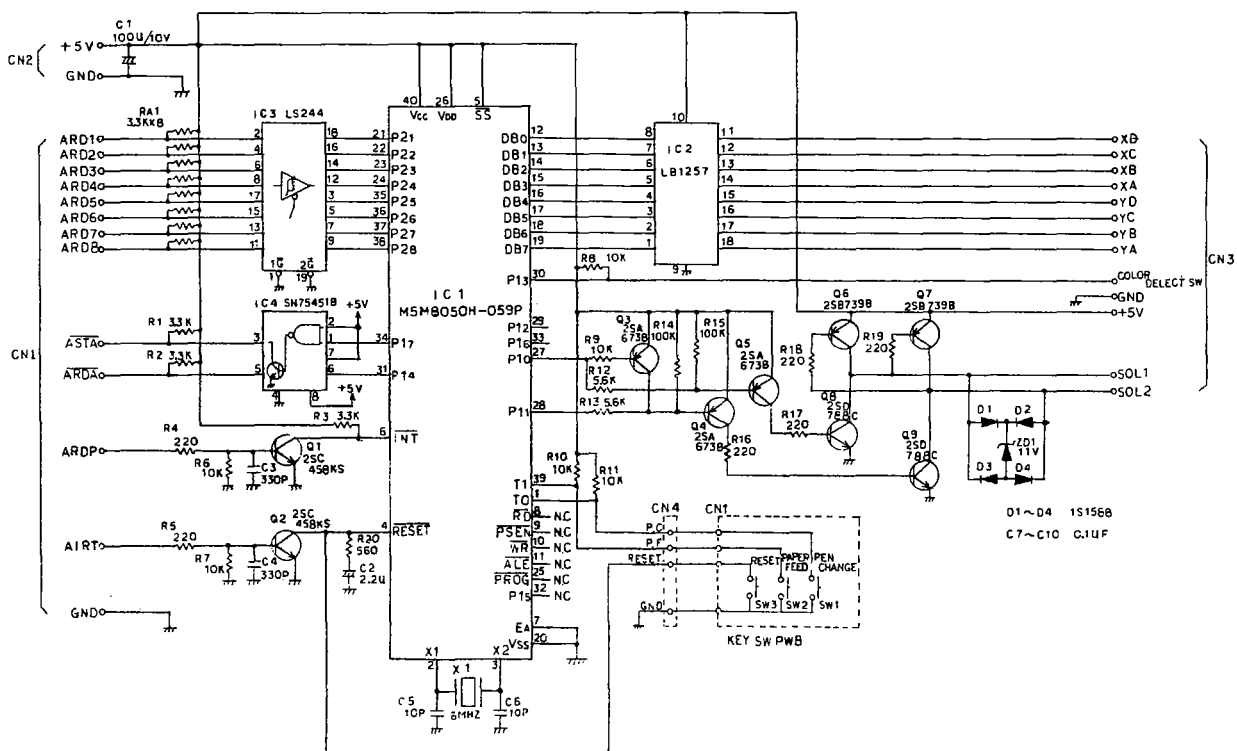
Outline

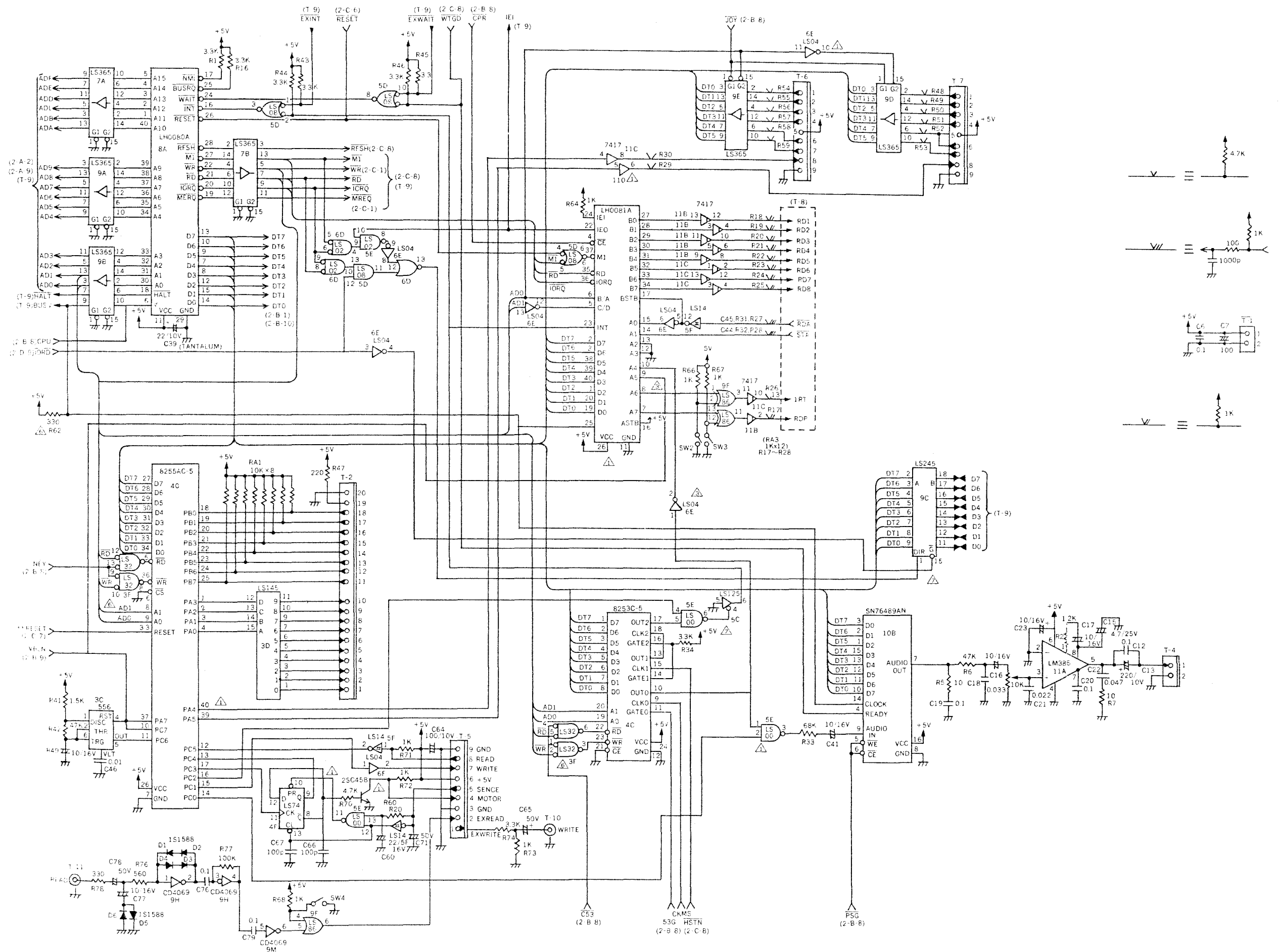
The MZ-1P16 is the external installation 4-color plot printer designed for use with the MZ-800 series personal computer. It can be fixed on the MZ-800 when the table is used.

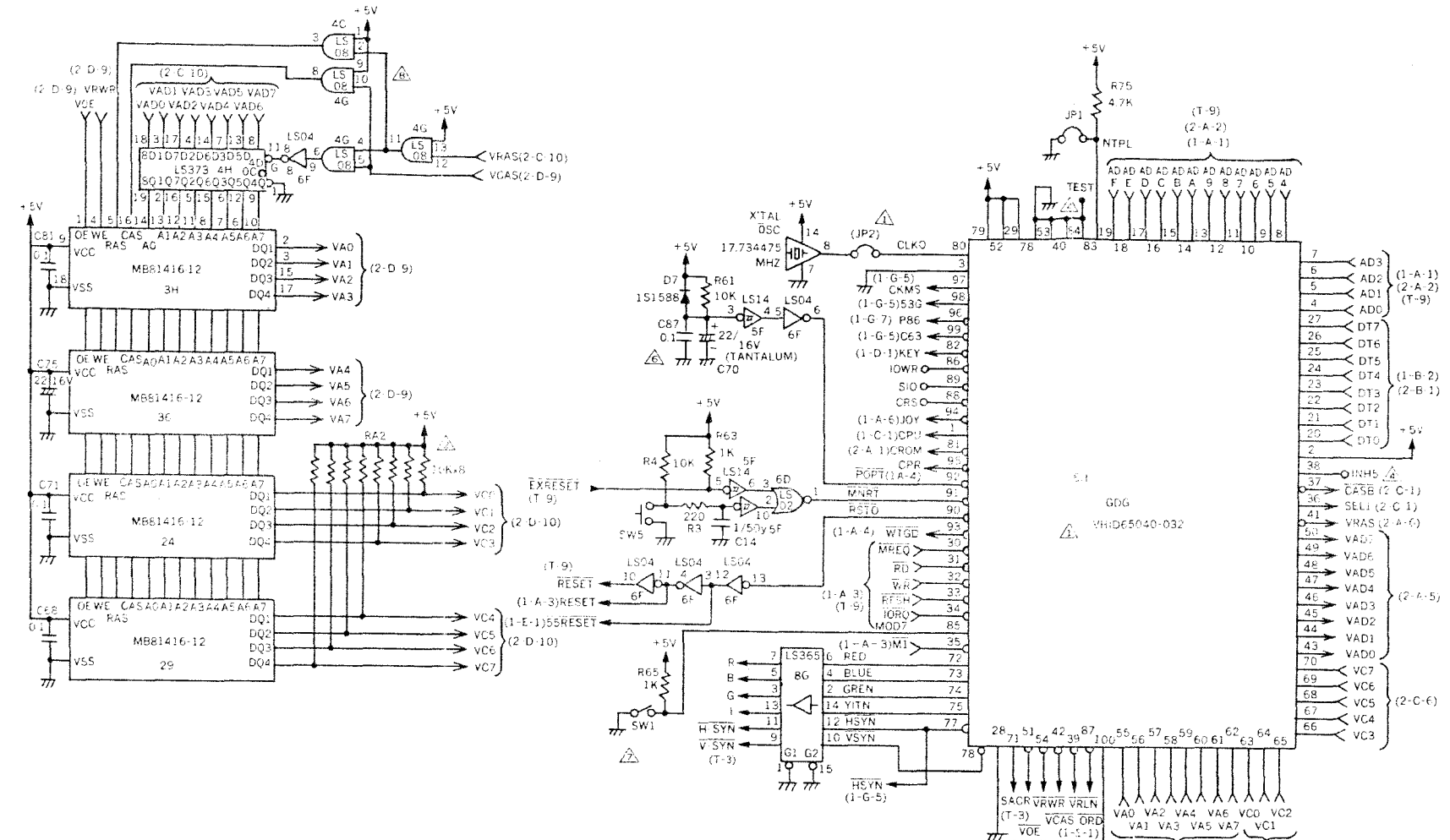
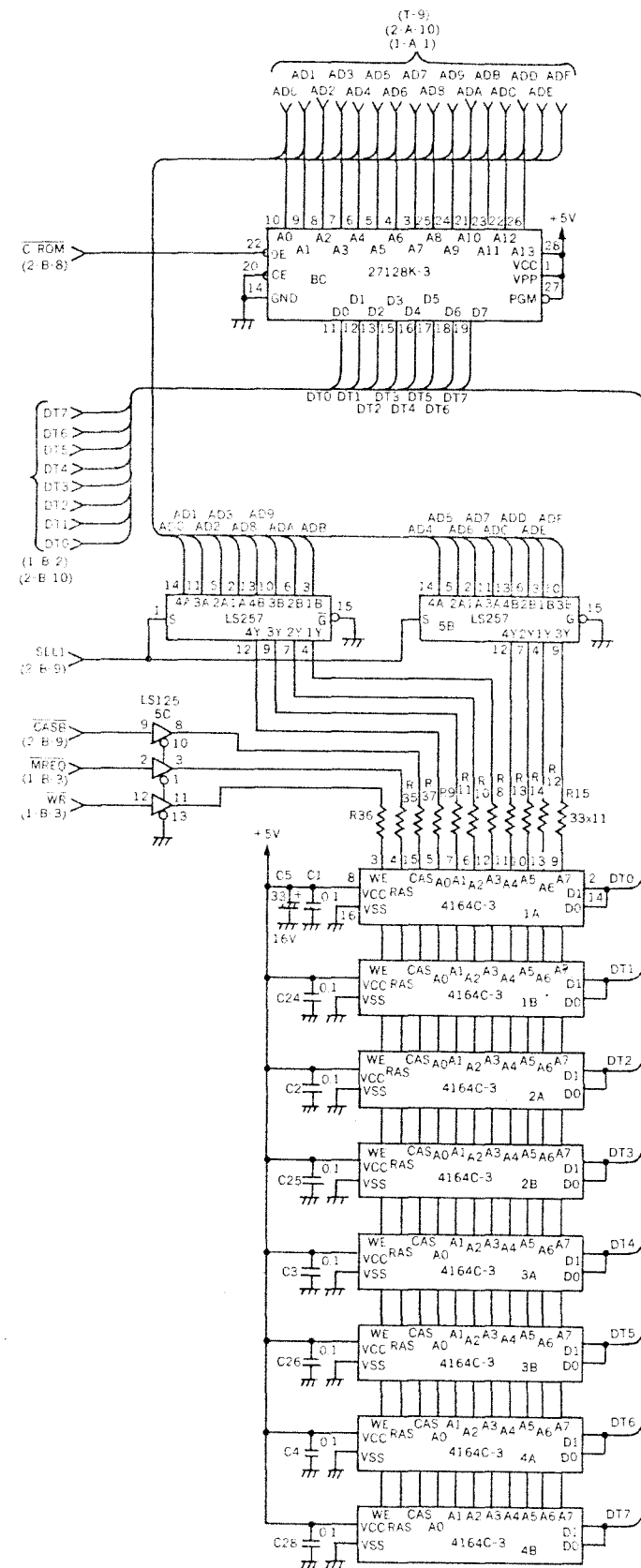
Specification

Type name	: MZ1P16
Print method	: 1, Black; 2, Blue; 3, Green; 4, Red
Print speed	: 10 characters average (smallest letter)
Printing size	: 80/40/26 digits (software assigned)
Character set	: 115
Resolution	: 0.2 mm
Power supply	: +5 V supplied from the MZ-800 via the DC jack
Power consumption	: 11 W
Physical dimensions	: 162(W) × 133(D) × 59(H), excluding accessories.
Weight	: 1 kg (MZ-1P16)
Accessories	: Roll paper (1), ball point pen (one each of black, blue, green, red), paper holder (one each on side), paper shaft (1), paper guide (1)
Operating temperature	: 0 to +35 centigrades
Storage temperature	: -20 to +70 centigrades
Operating humidity	: 80%RH, maximum

11-16 Circuit







T-9	
PARTS SIDE	SOLDER SIDE
1 +5V	2 +5V
3 D2	4 D3
5 D1	6 D4
7 D0	8 D5
9 GND	10 D6
11 ADF	12 D7
13 ADE	14 BUS
15 ADD	16 MI
17 ADC	18 WR
19 ADB	20 RD
21 ADA	22 IORQ
23 AD9	24 MREQ
25 AD8	26 GND
27 AD7	28 HALT
29 AD6	30 IEI Δ
31 AD5	32 NC
33 AD4	34 RESET
35 AD3	36 EXRESET
37 AD2	38 EXINT
39 AD1	40 EXWAIT
41 AD0	42 NC
43 GND	44 GND

T-6	
PARTS SIDE	SOLDER SIDE
1 JOY STICK A	
2 FWDA	
3 BACKA	
4 LEFTA	
5 RIGHTA	
6 +5V	
7 TRG 1 A	
8 TRG 2 A	
9 COMA	
10 GND	

T-3	
PARTS SIDE	SOLDER SIDE
1 GND	
2 I	
3 GND	
4 H-SYNC	
5 V-SYNC	
6 GND	
7 +5V	
8 G	
9 B	
10 R	
11 SBCR	
12 GND	

T-8	
PARTS SIDE	SOLDER SIDE
1 RDP	2 GND
3 RD 1	4 GND
5 RD 2	6 GND
7 RD 3	8 GND
9 RD 4	10 GND
11 RD 5	12 GND
13 RD 6	14 GND
15 RD 7	16 GND
17 RD 8	18 GND
19 IRT	20 GND
21 RDA	22 GND
23 STA	24 GND
25 GND	26 GND

T-7	
PARTS SIDE	SOLDER SIDE
1 JOY STICK B	
2 FWDB	
3 LEFTB	
4 RIGHTB	
5 +5V	
6 TRG 1 B	
7 TRG 2 B	
8 COMB	
9 GND	
10 GND	
11 SW 1	
12 SW 2	
13 SW 3	
14 SW 4	
15 +5V	

SOFTWARE

1 Character generator (C.G.)

1-1 How to modify C.G. ROM

We will explain here the FONT address in the character generator ROM. The display code corresponding to the character is referenced from the display code table, and multiplied by eight. The result after adding 1000H becomes the address in the ROM where the FONT is stored. When the target character is in the display code table (second 2K-byte half), that display code is multiplied by eight and 1800H is added.

(Example)

The display code for the “☹” character is CFH.

Therefore,

$$\text{CFH} \times 8 + 1000\text{H} = 1678\text{H}$$

The address of the FONT pattern is, therefore, 1678H.

Looking at the character generator dump list we see:

C.G. address	DATA	FONT																																																																
1678H	3CH	<table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	1	1	1	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0																																																											
0	1	0	0	0	0	1	0																																																											
1	0	0	0	0	0	0	1																																																											
1	0	1	0	0	1	0	1																																																											
1	0	0	0	0	0	0	1																																																											
1	0	0	1	1	0	0	1																																																											
0	1	0	0	0	0	1	0																																																											
0	0	1	1	1	1	0	0																																																											
1679H	42H																																																																	
167AH	81H																																																																	
167BH	A5H																																																																	
167CH	81H																																																																	
167DH	99H																																																																	
167EH	42H																																																																	
167FH	3CH																																																																	

↑

LSD

↑

MSD

1-2 C.G.Table

■ Display code table

The display codes are used to address character patterns stored in the character generator. These codes must be transferred to video-RAM to display characters.

Monitor subroutines PRNT(0012H) and MSG(0015H) convert ASCII codes into display codes and transfer them to the V-RAM location indicated for the cursor.

Codes CIH to C6H are for controlling the cursor.

MSD LSD		0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0	0 0 0 0	SP	P	0	□	}	↑	π	□		p	□	□	↓	□	□	SP
1	0 0 0 1	A	Q	I	□	♠	<	!	□	a	q	□	□	↓	□	□	□
2	0 0 1 0	B	R	2	□	□	□	"	□	b	r	□	□	↑	□	□	□
3	0 0 1 1	C	S	3	□	■	♥	#	□	c	s	□	□	→	□	□	□
4	0 1 0 0	D	T	4	□	♦	□	\$	□	d	t	□	□	←	□	□	□
5	0 1 0 1	E	U	5	□	←	@	%	□	e	u	□	□	□	□	□	□
6	0 1 1 0	F	V	6	□	♣	□	&	□	f	v	□	□	□	□	□	□
7	0 1 1 1	G	W	7	□	●	>	'	□	g	w	□	□	□	□	□	□
8	1 0 0 0	H	X	8	□	○	↓	(	□	h	x	□	□	□	□	□	□
9	1 0 0 1	I	Y	9	□	?	□	)	□	i	y	□	□	□	□	□	□
A	1 0 1 0	J	Z	—	□	□	→	+	□	j	z	□	□	□	□	□	□
B	1 0 1 1	K	£	=	□	□	□	*	□	k	ä	ü	□	□	□	□	□
C	1 1 0 0	L	□	;	□	□	□	□	□	l	□	ö	□	□	□	□	□
D	1 1 0 1	M	□	/	□	□	□	□	□	m	□	ü	□	□	□	□	□
E	1 1 1 0	N	□	.	□	□	□	□	□	n	□	Ä	□	□	□	□	□
F	1 1 1 1	O	□	,	□	□	□	□	□	o	□	Ö	□	□	□	□	□

■MZ-700 Display code table (second 2K-byte half)

MSD \ LSD		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0 0 0 0																
1	0 0 0 1																
2	0 0 1 0																
3	0 0 1 1																
4	0 1 0 0																
5	0 1 0 1																
6	0 1 1 0																
7	0 1 1 1																
8	1 0 0 0																
9	1 0 0 1																
A	1 0 1 0																
B	1 0 1 1																
C	1 1 0 0																
D	1 1 0 1																
E	1 1 1 0																
F	1 1 1 1																

1-3 C.G. ROM dump list

(address)	(data)
0000	00 00 00 00 00 00 00 00 18 24 42 7E 42 42 42 00
0010	3E 44 44 3C 44 44 3E 00 38 44 02 02 02 44 38 00
0020	1E 24 44 44 44 24 1E 00 7E 02 02 1E 02 02 7E 00
0030	7E 02 02 1E 02 02 02 00 38 44 02 72 42 44 38 00
0040	42 42 42 7E 42 42 42 00 38 10 10 10 10 10 38 00
0050	70 20 20 20 20 22 1C 00 42 22 12 0E 12 22 42 00
0060	02 02 02 02 02 02 7E 00 42 66 5A 5A 42 42 42 00
0070	42 46 4A 52 62 42 42 00 18 24 42 42 42 24 18 00
0080	3E 42 42 3E 02 02 02 00 18 24 42 42 52 24 58 00
0090	3E 42 42 3E 12 22 42 00 3C 42 02 3C 40 42 3C 00
00A0	7C 10 10 10 10 10 10 00 42 42 42 42 42 42 3C 00
00B0	42 42 42 24 24 18 18 00 42 42 42 5A 5A 66 42 00
00C0	42 42 24 18 24 42 42 00 44 44 44 38 10 10 10 00
00D0	7E 40 20 18 04 02 7E 00 30 48 08 1C 08 08 7C 00
00E0	10 10 10 10 F0 00 00 00 10 10 10 10 1F 00 00 00
00F0	10 10 10 10 F0 10 10 10 10 10 10 10 FF 00 00 00
0100	3C 42 62 5A 46 42 3C 00 10 18 14 10 10 10 7C 00
0110	3C 42 40 3C 0C 02 7E 00 3C 42 40 3C 40 42 3C 00
0120	20 30 28 24 7E 20 20 00 7E 02 1E 20 40 22 1C 00
0130	38 04 02 3E 42 42 3C 00 7E 42 20 10 08 08 08 00
0140	3C 42 42 3C 42 42 3C 00 3C 42 42 7C 40 20 1C 00
0150	00 00 00 7E 00 00 00 00 00 00 7E 00 7E 00 00 00
0160	00 00 10 00 00 10 10 08 00 40 20 10 08 04 02 00
0170	00 00 00 00 00 18 18 00 00 00 00 00 00 10 10 08
0180	00 FF 00 00 00 00 00 00 02 02 02 02 02 02 02 02
0190	01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80 FF
01A0	00 00 00 FF 00 00 00 00 08 08 08 08 08 08 08 08
01B0	FF FF 00 00 00 00 00 00 03 03 03 03 03 03 03 03
01C0	00 00 00 00 00 FF 00 00 20 20 20 20 20 20 20 20
01D0	00 00 00 00 FF FF FF FF F0 F0 F0 F0 F0 F0 F0 F0
01E0	00 00 00 00 00 00 00 00 FF 80 80 80 80 80 80 80 80
01F0	00 00 00 00 00 00 FF FF C0 C0 C0 C0 C0 C0 C0 C0
0200	08 10 10 20 10 10 08 00 10 38 7C FE FE 38 7C 00
0210	FF FE FC F8 F0 E0 C0 80 FF FF FF FF FF FF FF FF
0220	10 38 7C FE 7C 38 10 00 00 00 08 04 FE 04 08 00
0230	38 38 D6 FE D6 10 38 00 00 3C 7E 7E 7E 7E 3C 00
0240	00 3C 42 42 42 42 3C 00 3C 42 40 30 08 00 08 00
0250	FF C3 81 81 81 81 C3 FF 00 00 00 00 C0 20 10 10
0260	00 00 00 00 03 04 08 08 01 03 07 0F 1F 3F 7F FF
0270	80 C0 E0 F0 F8 FC FE FF 00 00 10 00 00 10 00 00
0280	00 10 38 54 10 10 10 00 70 18 0C 06 0C 18 70 00
0290	3C 04 04 04 04 04 3C 00 6C FE FE FE 7C 38 10 00
02A0	3C 20 20 20 20 20 3C 00 38 44 52 6A 32 04 78 00
02B0	FF 7F 3F 1F 0F 07 03 01 0E 18 30 60 30 18 0E 00
02C0	00 10 10 10 54 38 10 00 00 02 04 08 10 20 40 00
02D0	00 00 20 40 FE 40 20 00 0F 0F 0F 0F F0 F0 F0 F0
02E0	00 00 00 00 F0 10 10 10 00 00 00 00 1F 10 10 10
02F0	10 10 10 10 1F 10 10 10 00 00 00 00 FF 10 10 10
0300	00 00 80 7C 2A 28 28 00 10 10 10 10 00 00 10 00
0310	24 24 24 00 00 00 00 00 24 24 7E 24 7E 24 24 00
0320	10 78 14 38 50 38 10 00 00 46 26 10 08 64 62 00
0330	0C 12 12 0C 52 22 5C 00 20 10 08 00 00 00 00 00
0340	20 10 08 08 08 10 20 00 04 08 10 10 10 08 04 00
0350	00 10 10 7C 10 10 00 00 10 54 38 7C 38 54 10 00
0360	F0 F0 F0 F0 0F 0F 0F 0F 91 42 24 18 18 24 42 81
0370	08 08 04 03 00 00 00 00 10 10 20 C0 00 00 00 00
0380	FF 00 00 00 00 00 00 00 01 01 01 01 01 01 01 01
0390	FF 01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80
03A0	00 00 FF 00 00 00 00 00 04 04 04 04 04 04 04 04
03B0	80 40 20 10 08 04 02 01 01 02 04 08 10 20 40 80
03C0	00 00 00 00 FF 00 00 00 10 10 10 10 10 10 10 10
03D0	FF FF FF FF 00 00 00 00 0F 0F 0F 0F 0F 0F 0F 0F
03E0	00 00 00 00 00 00 00 FF 00 40 40 40 40 40 40 40
03F0	00 00 00 00 00 FF FF FF E0 E0 E0 50 E0 E0 E0 E0
0400	18 18 18 18 18 18 18 00 00 00 1C 20 3C 22 5C 00
0410	02 02 3A 46 42 46 3A 00 00 00 3C 42 02 42 3C 00
0420	40 40 5C 62 42 62 5C 00 00 00 3C 42 7E 02 3C 00
0430	30 48 08 3E 08 08 08 00 00 00 5C 62 62 5C 40 3C
0440	02 02 3A 46 42 42 42 00 10 00 18 10 10 10 38 00

1-3 C.G. ROM dump list

(address)	(data)
0000	00 00 00 00 00 00 00 00 18 24 42 7E 42 42 42 00
0010	3E 44 44 3C 44 44 3E 00 38 44 02 02 02 44 38 00
0020	1E 24 44 44 44 24 1E 00 7E 02 02 1E 02 02 7E 00
0030	7E 02 02 1E 02 02 02 00 38 44 02 72 42 44 38 00
0040	42 42 42 7E 42 42 42 00 38 10 10 10 10 10 38 00
0050	70 20 20 20 20 22 1C 00 42 22 12 0E 12 22 42 00
0060	02 02 02 02 02 02 7E 00 42 66 5A 5A 42 42 42 00
0070	42 46 4A 52 62 42 42 00 18 24 42 42 42 24 18 00
0080	3E 42 42 3E 02 02 02 00 18 24 42 42 52 24 58 00
0090	3E 42 42 3E 12 22 42 00 3C 42 02 3C 40 42 3C 00
00A0	7C 10 10 10 10 10 10 00 42 42 42 42 42 42 3C 00
00B0	42 42 42 24 24 18 18 00 42 42 42 5A 5A 66 42 00
00C0	42 42 24 18 24 42 42 00 44 44 44 38 10 10 10 00
00D0	7E 40 20 18 04 02 7E 00 30 48 08 1C 08 08 7C 00
00E0	10 10 10 10 F0 00 00 00 10 10 10 10 1F 00 00 00
00F0	10 10 10 10 F0 10 10 10 10 10 10 10 FF 00 00 00
0100	3C 42 62 5A 46 42 3C 00 10 18 14 10 10 10 7C 00
0110	3C 42 40 3C 0C 02 7E 00 3C 42 40 3C 40 42 3C 00
0120	20 30 28 24 7E 20 20 00 7E 02 1E 20 40 22 1C 00
0130	38 04 02 3E 42 42 3C 00 7E 42 20 10 08 08 08 00
0140	3C 42 42 3C 42 42 3C 00 3C 42 42 7C 40 20 1C 00
0150	00 00 00 7E 00 00 00 00 00 00 7E 00 7E 00 00 00
0160	00 00 10 00 00 10 10 08 00 40 20 10 08 04 02 00
0170	00 00 00 00 00 18 18 00 00 00 00 00 00 10 10 08
0180	00 FF 00 00 00 00 00 00 02 02 02 02 02 02 02 02
0190	01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80 FF
01A0	00 00 00 FF 00 00 00 00 08 08 08 08 08 08 08 08
01B0	FF FF 00 00 00 00 00 00 03 03 03 03 03 03 03 03
01C0	00 00 00 00 00 FF 00 00 20 20 20 20 20 20 20 20
01D0	00 00 00 00 FF FF FF FF F0 F0 F0 F0 F0 F0 F0 F0
01E0	00 00 00 00 00 00 00 00 FF 80 80 80 80 80 80 80 80
01F0	00 00 00 00 00 00 FF FF C0 C0 C0 C0 C0 C0 C0 C0
0200	08 10 10 20 10 10 08 00 10 38 7C FE FE 38 7C 00
0210	FF FE FC F8 F0 E0 C0 80 FF FF FF FF FF FF FF FF
0220	10 38 7C FE 7C 38 10 00 00 00 08 04 FE 04 08 00
0230	38 38 D6 FE D6 10 38 00 00 3C 7E 7E 7E 7E 3C 00
0240	00 3C 42 42 42 42 3C 00 3C 42 40 30 08 00 08 00
0250	FF C3 81 81 81 81 C3 FF 00 00 00 00 C0 20 10 10
0260	00 00 00 00 03 04 08 08 01 03 07 0F 1F 3F 7F FF
0270	80 C0 E0 F0 F8 FC FE FF 00 00 10 00 00 10 00 00
0280	00 10 38 54 10 10 10 00 70 18 0C 06 0C 18 70 00
0290	3C 04 04 04 04 04 3C 00 6C FE FE FE 7C 38 10 00
02A0	3C 20 20 20 20 20 3C 00 38 44 52 6A 32 04 78 00
02B0	FF 7F 3F 1F 0F 07 03 01 0E 18 30 60 30 18 0E 00
02C0	00 10 10 10 54 38 10 00 00 02 04 08 10 20 40 00
02D0	00 00 20 40 FE 40 20 00 0F 0F 0F 0F F0 F0 F0 F0
02E0	00 00 00 00 F0 10 10 10 00 00 00 00 1F 10 10 10
02F0	10 10 10 10 1F 10 10 10 00 00 00 00 FF 10 10 10
0300	00 00 80 7C 2A 28 28 00 10 10 10 10 00 00 10 00
0310	24 24 24 00 00 00 00 00 24 24 7E 24 7E 24 24 00
0320	10 78 14 38 50 38 10 00 00 46 26 10 08 64 62 00
0330	0C 12 12 0C 52 22 5C 00 20 10 08 00 00 00 00 00
0340	20 10 08 08 08 10 20 00 04 08 10 10 10 08 04 00
0350	00 10 10 7C 10 10 00 00 10 54 38 7C 38 54 10 00
0360	F0 F0 F0 F0 0F 0F 0F 0F 91 42 24 18 18 24 42 81
0370	08 08 04 03 00 00 00 00 10 10 20 C0 00 00 00 00
0380	FF 00 00 00 00 00 00 00 01 01 01 01 01 01 01 01
0390	FF 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80
03A0	00 00 FF 00 00 00 00 00 04 04 04 04 04 04 04 04
03B0	80 40 20 10 08 04 02 01 01 02 04 08 10 20 40 80
03C0	00 00 00 00 FF 00 00 00 10 10 10 10 10 10 10 10
03D0	FF FF FF FF 00 00 00 00 0F 0F 0F 0F 0F 0F 0F 0F
03E0	00 00 00 00 00 00 00 FF 00 40 40 40 40 40 40 40
03F0	00 00 00 00 00 FF FF FF E0 E0 E0 50 E0 E0 E0 E0
0400	18 18 18 18 18 18 18 00 00 00 1C 20 3C 22 5C 00
0410	02 02 3A 46 42 46 3A 00 00 00 3C 42 02 42 3C 00
0420	40 40 5C 62 42 62 5C 00 00 00 3C 42 7E 02 3C 00
0430	30 48 08 3E 08 08 08 00 00 00 5C 62 62 5C 40 3C
0440	02 02 3A 46 42 42 42 00 10 00 18 10 10 10 38 00

(address)	(data)
08A0	7F 41 77 14 14 14 14 1C 77 55 55 55 55 5D 63 3E
08B0	77 55 55 55 55 2A 14 08 77 55 55 5D 55 55 41 7F
08C0	63 55 2A 14 14 2A 55 63 77 55 49 22 14 14 14 1C
08D0	7F 41 5F 28 14 7A 41 7F 00 02 05 09 FF 7E 00 00
08E0	00 40 A0 90 FF 7E 00 00 00 3E 68 3E 1C 2A 49 00
08F0	49 2A 1C 7F 1C 2A 49 00 00 00 1C 2A 7F 00 00 00
0900	3E 41 4D 55 55 59 41 3E 1C 12 16 14 14 36 22 3E
0910	3E 41 5D 53 28 74 42 7F 3F 41 5F 44 44 5F 41 3F
0920	30 28 24 2A 6D 41 6F 38 7F 41 7D 21 5E 5F 61 3F
0930	7E 41 7D 3D 41 5D 41 3E 7F 41 5F 28 14 0A 0A 0E
0940	3E 41 5D 3E 41 5D 41 3E 3E 41 5D 41 5E 5F 41 3F
0950	1F 11 7D 55 5F 44 7C 00 FB 88 BE AA FA 22 3E 00
0960	3C 5A FF E7 7E 24 42 81 3C 5A FF E7 7E 24 24 66
0970	10 38 54 FE EE 7C 6C 06 10 38 54 FE EE 7C 6C 28
0980	82 45 3C 5A 7E FF 42 06 41 A2 3C 5A 7E FF 42 63
0990	00 5A BD 99 24 42 24 00 81 A5 5A 18 18 24 03 00
09A0	00 24 7E BD 7E 24 24 E7 24 7E BD 7E 24 42 42 03
09B0	3C 5A FF D5 AB FF BB 91 3C 5A FF D5 AB FF EE 44
09C0	3C 42 A5 81 99 81 AB 55 3C 42 A5 81 99 81 D5 AA
09D0	42 42 66 E7 FF FF 7E 3C 38 7F FC F0 F0 FC 7F 38
09E0	3C 7E FF FF E7 66 42 42 1C FE 3F 0F 0F 3F FE 1C
09F0	3C 7E FF FF FF FF 7E 3C 08 1C 14 14 14 3E 7F 6B
0A00	00 00 E0 7E E3 7E E0 00 06 FE 7C 28 28 28 38 10
0A10	00 03 07 7E 07 7E 07 03 3C 30 3C 18 3C 6E 6E 62
0A20	3C 24 3C 18 3C 5A 5A 7E 3C 0C 3C 18 3C 76 76 46
0A30	7E 7E 24 24 24 24 24 36 7E 7E 24 24 24 24 66
0A40	7E 7E 24 24 24 24 24 6C 44 06 EF ED FF 7E 3C 3C
0A50	1C 36 FF FC F0 FC 3F 1C 3C 3C 7E FF B7 F7 63 22
0A60	38 6C FF 3F 0F FC 78 3C 7E FF FD FF 7E 3C 3C
0A70	3C 3C 7E FF BF FF 7E 3C 38 6C FF FF FF 7C 38
0A80	1C 36 FF FF FF FF 3E 1C 18 3C 3C 3C 18 3C 3C
0A90	00 00 DE FF FF DE 00 00 3C 3C 18 3C 3C 3C 18
0AA0	00 00 7B FF FF 7B 00 00 04 06 04 04 0C 14 3C 3C
0AB0	00 02 FF D0 E0 00 00 00 3C 28 30 20 20 60 20
0AC0	00 40 FF 0B 07 03 00 00 08 08 1C 3E 49 08 08 1C
0AD0	00 10 08 8C FF 8C 08 10 1C 08 08 49 3E 1C 08 08
0AE0	00 08 10 31 FF 31 10 08 00 1E 06 0A 12 20 40 00
0AF0	00 40 20 12 0A 06 1E 00 00 02 04 48 50 60 78 00
0B00	00 78 60 50 48 04 02 00 18 7E 7E FF C3 81 81 81
0B10	F8 1E 0E 0F 0F 0E 1E F8 81 81 81 C3 5F 7E 7E 18
0B20	1F 78 70 F0 F0 70 78 1F F0 85 B5 A5 A5 BD 81 FF
0B30	FF 81 BD A5 A1 BF 80 FF FF 81 BD A5 A5 AD A1 BF
0B40	FF 01 FD 85 A5 BD 81 FF 00 18 00 3C 00 7E 00 FF
0B50	80 A0 A8 AA AA A8 A0 80 FF 00 7E 00 3C 00 18 00
0B60	01 05 15 55 55 15 05 01 00 10 38 7C 00 10 38 7C
0B70	00 00 88 CC EE CC 88 00 00 7C 38 10 00 7C 38 10
0B80	00 00 22 66 EE 66 22 00 00 00 E7 A5 E7 00 00 00
0B90	08 1C 2A 08 08 2A 1C 08 00 00 24 42 FF 42 24 00
0BA0	FE 82 44 38 10 10 10 FE AA AA AA AA AA AA AA
0BB0	FF 00 FF 00 FF 00 FF 00 A5 42 A5 00 00 A5 42 A5
0BC0	24 42 81 00 00 81 42 24 FF 01 F9 05 05 05 05 05
0BD0	FF 80 A7 88 A8 88 A8 88 00 00 00 FF 05 F5 05 FF
0BE0	00 00 00 FF 82 82 AA FF 05 F9 01 FF 0C 0C 0C 1E
0BF0	88 87 80 FF 30 30 78 01 55 01 A9 01 F1 01 FF
0C00	80 95 80 8A 80 87 80 FF 3C 42 D5 AB 08 08 28 10
0C10	00 00 18 24 24 18 00 00 00 18 24 42 42 24 18 00
0C20	3C 42 81 81 81 81 42 3C 00 00 00 18 18 00 00 00
0C30	00 00 3C 3C 3C 3C 00 00 00 7E 7E 7E 7E 7E 00
0C40	3C 42 B9 85 85 B9 42 3C FF FF FF E7 E7 FF FF FF
0C50	FF FF C3 C3 C3 C3 FF FF FF 81 81 81 81 81 81 81
0C60	04 0C 04 04 FF 7E 3C 00 3C 42 81 FF FF 81 42 3C
0C70	3C 5A 99 99 99 99 5A 3C 3C 5A 99 FF FF 99 5A 3C
0C80	00 14 7F 55 7F 2A 1C 08 F0 0C 02 72 51 71 01 81
0C90	0F 30 40 4E 8A 8E 80 81 F0 0C 02 02 71 01 01 81
0CA0	0F 30 40 40 8E 80 80 81 81 01 11 21 C2 02 0C F0
0CB0	81 80 88 84 43 40 30 0F 81 01 01 E1 02 02 0C F0
0CC0	81 80 80 87 40 40 30 0F 81 01 01 21 C2 02 0C F0
0CD0	81 80 83 84 43 40 30 0F 81 01 01 E1 11 12 02 0C F0
0CE0	81 80 87 88 48 40 30 0F 10 08 2A 7F 7F 7F 7F 3E

(address)	(data)
0CF0	00 60 10 08 0C 1E 1E 0C 00 4A 2C 60 96 34 52 00
0D00	89 4A 00 C0 03 00 52 91 01 03 07 0F FF FF FF FF
0D10	00 00 80 40 FF C3 C3 FF 00 00 01 02 FF C3 C3 FF
0D20	00 03 04 08 3F 7F FF 3F 80 C0 E0 F0 FF FF FF FF
0D30	40 28 14 10 28 28 10 00 00 7F 42 04 08 04 42 7F
0D40	00 C0 20 10 FC FE FF FC 00 04 08 08 08 14 12 61
0D50	00 3C 42 42 42 24 A5 E7 00 22 41 41 49 36 00 00
0D60	00 00 36 49 49 36 00 00 00 40 36 09 09 76 00 00
0D70	00 78 08 0A 0A 0D 08 00 00 00 08 00 3E 00 08 00
0D80	00 8F DA AA AA 8A 8A 00 FF 91 89 A3 C5 91 89 FF
0D90	FF C3 A5 99 99 A5 C3 FF 00 49 2A 1C 77 1C 2A 49
0DA0	FF 99 99 FF FF 99 99 FF 49 2A 1C 08 08 08 08 08
0DB0	1C 08 1C 08 1C 08 1C 08 00 00 00 55 FF 55 00 00
0DC0	00 08 08 3E 08 08 00 3E 7E 42 7E 42 7E 42 7E 42
0DD0	00 FF AA AA AA AA FF 00 00 00 00 03 0D 31 C1 FF
0DE0	00 00 00 C0 B0 8C 83 FF 00 00 00 00 3C 7E FF FF
0DF0	FF FF 7E 3C 00 00 00 00 03 07 0F 0F 0F 0F 07 03
0E00	C0 E0 F0 F0 F0 F0 E0 C0 C0 30 FC FC FF FE EC FB
0E10	03 0C 1D 3B 77 6F DF DF 70 70 50 20 80 80 C0 F0
0E20	5E 2E 2F 2F 2F 5F BF BF 20 72 27 62 F6 FE 06 FC
0E30	04 4E E4 46 6F 7F 60 3F DC 8C D8 F8 08 F8 F0 E0
0E40	3B 31 1B 1F 10 1F 0F 07 80 C0 E0 60 70 7C 0E 0C
0E50	01 03 07 06 0E 3E 70 30 78 70 60 E0 C0 EC FE D1
0E60	1E 0E 06 07 03 37 7F 8B 80 CC DE 9A 31 FB FE FC
0E70	01 33 7B 59 8C 0F 7F 3F FC F8 F8 F0 F0 FE 00 FF
0E80	3F 1F 1F 0F 0F 7F 00 FF 00 80 40 20 40 80 FB FB
0E90	00 01 02 04 02 01 1F 1F 40 40 40 40 FB 04 FE 00
0EA0	02 02 02 02 1F 20 7F 00 CE CE CE FE FC FB F0 F0
0EB0	73 73 73 7F 3F 1F 0F 0F F0 F0 F0 18 FE 02 FE FF
0EC0	0F 0F 0F 1B 7F 40 7F FF 1F 22 42 84 84 42 22 1F
0ED0	FF A0 E0 00 00 E0 A0 FF 3F 61 41 81 81 41 61 3F
0EE0	00 00 01 02 FE 01 00 00 00 00 00 00 FF 80 80 80
0EF0	80 80 80 80 FF 00 00 00 FF 01 01 01 01 00 00 00
0F00	00 00 00 00 01 01 01 FF 00 10 30 50 9F 50 30 10
0F10	00 10 30 5C 97 5C 30 10 FB 14 12 7F 11 11 F1 00
0F20	02 03 02 67 90 40 20 F0 02 03 02 47 60 50 FB 40
0F30	02 03 02 F7 80 E0 80 F0 02 05 04 F2 87 E0 80 F0
0F40	03 06 18 60 18 06 01 7F 80 60 18 06 18 60 80 FE
0F50	00 80 60 B8 54 54 54 FB D8 F1 A6 88 93 95 8D CF
0F60	32 EF 0F 18 E0 40 7C 7F FE F9 8C 82 81 81 9F BF
0F70	11 40 02 00 11 82 00 89 02 80 11 00 02 20 01 88
0F80	00 0C 1A BF FF 9E 0C 00 00 30 58 FD FF 79 30 00
0F90	00 0C 1A BF FC 9F 0C 00 00 30 58 FD 3F F9 30 00
0FA0	08 14 16 3D 3F 1E 08 1C 5D 77 55 1C 1C 5D 7F 5D
0FB0	5D 7F 5D 1C 1C 55 77 5D 00 E7 42 FF F9 FF 42 E7
0FC0	00 E7 42 FF 9F FF 42 E7 00 00 3F 38 FE C6 7C 00
0FD0	00 00 FC 1C 7F 63 3E 00 FF 81 A5 81 81 A5 81 FF
0FE0	E7 81 81 00 00 81 81 E7 00 20 10 7F 08 7F 04 02
0FF0	1B 24 24 04 08 08 08 08 10 10 10 10 20 24 24 1B

MZ-800 monitor subroutines

The following subroutines are used by the ROM Monitor (9Z-504M). Each subroutine name symbolically represents the function of the corresponding subroutine. These subroutines can be called from user programs.

Registers saved are those whose contents are restored when control is returned to the calling program. The contents of other registers are changed by execution of the subroutine.

Name and entry point (hex.)	Function	Registers saved
CALL LETNL (0006)	Moves the cursor to the beginning of the next line.	All except AF
CALL PRNTS (000C)	Displays a space at the cursor position.	All except AF
CALL PRNTS (0012)	Displays the character corresponding to the ASCII code stored in the ACC at the cursor position. See Appendix J for the ASCII codes. No character is displayed when code 0D (carriage return) or codes 11 to 16 (the cursor control codes) are entered, but the corresponding function is still performed (a carriage return for 0D and cursor movement for 11 to 16).	All except AF
CALL MSG (0015)	Displays a message, starting at the position of the cursor. The starting address of the area in which the message is stored must be loaded into the DE register before calling this subroutine, and the message must end with a carriage return code (0D). The carriage return is not executed. The cursor is moved if any cursor control codes (11 to 16) are included in the message.	All registers
CALL BELL (003E)	Briefly sounds tone of Ia (about 880 Hz)	All except AF
CALL MELDY (0030)	Plays a tune according to the music data stored in the memory area starting at the address in the DE register. The music data must be in the same format as that for the MUSIC statement of the BASIC, and must end with 0D or C8. When the tune is completed, control is returned to the calling program with the C flag set to 0. When play is interrupted with the BREAK key, control is returned with the C flag set to 1.	All except AF
CALL XTEMP (0041)	Sets the music tempo according to the tempo data stored in the accumulator (ACC). ACC ← 01 Slowest speed ACC ← 04 Middle speed ACC ← 07 Highest speed Note that the data in the accumulator is not the ASCII codes for 1 to 7 but the binary codes.	All registers
CALL MSTA (0044)	Generates a continuous sound of the specified frequency. The frequency is given by the following equation $\text{freq.} = 895 \text{ kHz}/nn'$. Here, nn' is a 2-byte number stored in addresses 11A1 and 11A2 (n in 11A2 and n' in 11A1)	BC and DE

Name and entry point (hex.)	Function	Registers saved																																				
CALL MSTP (0047)	Stops the sound generated with the CALL MSTA subroutine.	All except AF																																				
CALL TIMST (0033)	Sets and starts the built-in clock. The registers must be set as follows before this routine is called. ACC ← 0 (AM), ACC ← 1 (PM) DE ← 4-digit hexadecimal number representing the time in seconds.	All except AF																																				
CALL TIMRD (003B)	Reads the built-in clock and returns the time as follows. ACC ← 0 (AM), ACC ← 1 (PM) DE ← 4-digit hexadecimal number representing the time in seconds.	All except AF and DE																																				
CALL BRKEY (001E)	Checks whether the SHIFT and BREAK keys are both being pressed. The Z flag is set when they are being pressed simultaneously; otherwise, it is reset.	All except AF																																				
CALL GETL (0003)	Reads one line of data from the keyboard and stores it in the memory area starting at the address in the DE register. This routine stops reading data when the CR key is pressed, then appends a carriage return code (0D) to the end of the data read. A maximum of 80 characters (including the carriage return code) can be entered in one line. Characters keyed in are echoed back to the display. Cursor control codes can be included in the line. When the SHIFT and BREAK keys are pressed simultaneously, the BREAK code is stored at the address indicated by the DE register and a carriage return code is stored in the following address.	All registers																																				
CALL GETKY (001B)	Reads a character code (ASCII) from the keyboard. If no key is pressed, control is returned to the calling program with 00 set in ACC. No provision is made to avoid data read errors due to key bounce, and characters entered are not echoed back to the display. When any of the special keys (such as DEL or CR) are pressed, this subroutine returns a code to the ACC which is different to the corresponding ASCII code as shown below. Here, display codes are used to address characters stored in the character generator, and are different from the ASCII codes. <table> <tr> <th>Special key</th><th>Code loaded in ACC</th><th>Display code</th></tr> <tr><td>DEL</td><td>60</td><td>C7</td></tr> <tr><td>INST</td><td>61</td><td>C8</td></tr> <tr><td>ALPHA</td><td>62</td><td>C9</td></tr> <tr><td>BREAK</td><td>64</td><td>CB</td></tr> <tr><td>CR</td><td>66</td><td>CD</td></tr> <tr><td>↓</td><td>11</td><td>C1</td></tr> <tr><td>↑</td><td>12</td><td>C2</td></tr> <tr><td>→</td><td>13</td><td>C3</td></tr> <tr><td>←</td><td>14</td><td>C4</td></tr> <tr><td>HOME</td><td>15</td><td>C5</td></tr> <tr><td>CLR</td><td>16</td><td>C6</td></tr> </table>	Special key	Code loaded in ACC	Display code	DEL	60	C7	INST	61	C8	ALPHA	62	C9	BREAK	64	CB	CR	66	CD	↓	11	C1	↑	12	C2	→	13	C3	←	14	C4	HOME	15	C5	CLR	16	C6	All except AF
Special key	Code loaded in ACC	Display code																																				
DEL	60	C7																																				
INST	61	C8																																				
ALPHA	62	C9																																				
BREAK	64	CB																																				
CR	66	CD																																				
↓	11	C1																																				
↑	12	C2																																				
→	13	C3																																				
←	14	C4																																				
HOME	15	C5																																				
CLR	16	C6																																				

Name and entry point (hex.)	Function	Registers saved																												
CALL ASC (03DA)	Loads the ASCII character corresponding to the hexadecimal number represented by the lower 4 bits of data in ACC.	All except AF																												
CALL HEX (03F9)	Converts the 8 data bits stored in the ACC into a hexadecimal number (assuming that the data is an ASCII character), then loads the hexadecimal number in the lower 4 bits of ACC. The C flag is set to 0 when a hexadecimal number is loaded in ACC; otherwise, it is set to 1.	All except AF																												
CALL HLHEX (0410)	Converts a string of 4 ASCII characters into a hexadecimal number and loads it in the HL register. The call and return conditions are as follows. DE ← Starting address of the memory area which contains the ASCII character string. (e.g., "3" "1" "A" "5") CALL HLHEX CF=0 HL ← hexadecimal number (e.g., HL=31A5 _H) CF=1 The contents of HL are not guaranteed.	All except AF and HL																												
CALL 2HEX (041F)	Converts a string of 2 ASCII characters into a hexadecimal number and loads it into the ACC. The call and return conditions are as follows. DE ← Starting address of the memory area which contains the ASCII character string. (e.g., "3" "A") CALL 2HEX CF=0 ACC ← hexadecimal number (e.g., ACC=3A _H) CF=1 The contents of the ACC are not guaranteed.	All except AF and DE																												
CALL ??KEY (09B3)	Blinks the cursor to prompt for key input. When a key is pressed, the corresponding display code is loaded into the ACC and control is returned to the calling program.	All except AF																												
CALL ?ADCN (0BB9)	Converts ASCII codes into display codes. The call and return conditions are as follows. ACC ← ASCII code CALL ?ADCN ACC ← Display code	All except AF																												
CALL ?DACN (0BCE)	Converts display codes into ASCII codes. The call and return conditions are as follows. ACC ← Display codes CALL ?DACN ACC ← ASCII code	All except AF																												
CALL ?BLNK (0DA6)	Detects the vertical blanking period. Control is returned to the calling program when the vertical blanking period is entered.	All registers																												
CALL ?DPCT (0DDC)	Controls display as follows. <table><tr><th>ACC</th><th>Control</th><th>ACC</th><th>Control</th></tr><tr><td>C0_H</td><td>Scrolling</td><td>C6_H</td><td>Same as the CLR key.</td></tr><tr><td>C1_H</td><td>Same as the ↓ key.</td><td>C7_H</td><td>Same as the DEL key.</td></tr><tr><td>C2_H</td><td>Same as the ↑ key.</td><td>C8_H</td><td>Same as the INST key.</td></tr><tr><td>C3_H</td><td>Same as the → key.</td><td>C9_H</td><td>Same as the ALPHA key.</td></tr><tr><td>C4_H</td><td>Same as the ← key.</td><td>CD_H</td><td>Same as the CR key.</td></tr><tr><td>C5_H</td><td>Same as the HOME key.</td><td></td><td></td></tr></table>	ACC	Control	ACC	Control	C0 _H	Scrolling	C6 _H	Same as the CLR key.	C1 _H	Same as the ↓ key.	C7 _H	Same as the DEL key.	C2 _H	Same as the ↑ key.	C8 _H	Same as the INST key.	C3 _H	Same as the → key.	C9 _H	Same as the ALPHA key.	C4 _H	Same as the ← key.	CD _H	Same as the CR key.	C5 _H	Same as the HOME key.			All registers
ACC	Control	ACC	Control																											
C0 _H	Scrolling	C6 _H	Same as the CLR key.																											
C1 _H	Same as the ↓ key.	C7 _H	Same as the DEL key.																											
C2 _H	Same as the ↑ key.	C8 _H	Same as the INST key.																											
C3 _H	Same as the → key.	C9 _H	Same as the ALPHA key.																											
C4 _H	Same as the ← key.	CD _H	Same as the CR key.																											
C5 _H	Same as the HOME key.																													
CALL ?POINT (0FB1)	Loads the current cursor location into the HL register. The return conditions are as follows. CALL ?POINT HL ← Cursor location (binary)	All except AF and HL																												

MZ-800 monitor call

Functions of this monitor can be called using their function numbers in the same manner as function calls. In the following explanation, two-digit hexadecimal numbers printed in Gothic are the function numbers and the characters at their right are the function names.

The table below lists the main monitor variables related to the monitor calls.

Monitor variable	Address in hexadecimal	Length in bytes	Function
SYSSTA	004D	2	Hot start address of the utilities using this monitor.
ERRORP	004F	2	Address of the error handling routine of the utilities using this monitor
ELMD	1000	1	File mode 1 : Object file 2 : BASIC text file 3 : Source file 4 : Relocatable file
ELMD1	1001	17	File name (up to 16 characters) and end mark ODH.
ELMD20	1014	2	File size in bytes
ELMD22	1016	2	Load address
ELMD24	1018	2	Execution address
ZLOG	1042	1	Logical number
ZRWX	1043	1	File open type 1 : Read open 2 : Write open
TEXTST	1070	2	Starting address of the text area of utilities using this monitor.
POOL	1072	2	Starting address of the work area of this monitor.
VARST	1074	2	Starting address of the variable area of utilities using this monitor.
TMPEND	107A	2	Ending address of the temporary area of utilities using this monitor.
TEMLMT	107E	2	Ending address of the memory area used by this monitor.
FILOUT	1091	1	Data is output to the CRT if the value at this address is zero and to the printer if it is 1. (The device specification is effective for monitor functions.& CR, .&IC, .&ICX and .&MSG)

00 .MONOP

Function: Returns to the RAM monitor.
Input registers: None
Output registers: None
Registers saved: None

01 .CR1

Function: Starts a new line independent of cursor position on a line
Input registers: None
Output registers: None
Registers saved: Primary registers only

02 .CR2

Function:	Starts a new line if the cursor is not at the beginning of a line.
Input registers:	None
Output registers:	None
Registers saved:	Primary register pairs only

03 .CRT1C

Function:	Outputs a character to the CRT. Control codes are executed.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

04 .CRT1X

Function:	Outputs a character to the CRT. Control codes are displayed in reverse video.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

05 .CRTMS

Function:	Outputs a character string. The end code is 00H. Control codes are executed. (Same as .CRT1C)
Input register:	DE: =Pointer position of the character string
Output registers:	None
Registers saved:	Primary register pairs only

06 .LPT0T

Function:	Outputs a character to the printer without code conversion.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

07 .LPT1C

Function:	Outputs a character to the printer converting its code into that of the printer used with the MZ computer. The PRINT/P statement of BASIC uses this function.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

08 .&CR

Function: Starts a new line on the CRT or printer according to the value of variable FILOUT(address 1091). Set the value of variable FILOUT to 1 to select the CRT or to 0 to select the printer in advance.

Input registers: None

Output registers: None

Registers saved: Primary register pairs only

09 .&1C

Function: Outputs a character to the CRT or printer according to the value of variable FILOUT. Control codes are executed when they are output to the CRT (same as .CRT1C). When this function outputs a character to the printer, its function is the same as .LPR1C.

Input register: ACC: =Output data

Output registers: None

Registers saved: Primary register pairs only

0A .&1CX

Function: Outputs a character to the CRT or printer according to the value of variable FILOUT. Control codes output to the CRT are displayed in reverse video (same as CRT1X). When the character is output to the printer, it is output in the same manner as with the PRINT/P statement of BASIC.

Input register: ACC: =Output data

Output registers: None

Registers saved: Primary register pairs only

0B .&MSG

Function: Outputs a character string to the CRT or printer. The end code is 00H. The switching condition between the CRT and printer is the same as that of the .&CRT. This function executes control codes when it outputs a character string to the CRT (same as .&CRT1C). When it outputs a character string to the printer, its function is the same as .LPT1C.

Input registers: DE: =Pointer position of the character string

Output registers: None

Registers saved: Primary register pairs only

0C .GETL

Function: Inputs one line of data from the keyboard and adds an end code 00H to the end of the data.

Input register: DE: =Starting address of the buffer in which the input data is stored.

Output register: CF: =1 when **SHIFT** + **BREAK** are depressed

Registers saved: Primary register pairs except AF

OD .INKEY

Function: Inputs a character from the keyboard.
Input registers: A: =0 Real time key scan (same as the GET statement of BASIC)
A: =1 Waits for key input blinking the cursor.
A: =FFH Unlike when A: =0, inputs only once if a key is depressed and held.
Output register: A: =MZ ASCII code
Registers saved: Primary register pairs except AF

OE .BREAK

Function: Detects **SHIFT** + **BREAK**.
Input register: None
Output registers: ZF: =1 when **SHIFT** + **BREAK** are pressed.
Registers saved: Primary register pairs except AF.

OF .HALT

Function: Waits for **SPACE** to be subsequently depressed if it is pressed. If **SHIFT** + **BREAK** are pressed next, this function transfers control to the address identified by .ERRORP.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

10 .DI

Function: Stops spooling or music, and inhibits interrupt.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

11 .EI

Function: Starts spooling or enables interrupt.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

17 .COUNT

Function: Counts the number of characters of the specified character string. The string must end with an end code 00H.
Input register: DE: =Pointer position of the character string
Output registers: ACC: =The length of the string
Registers saved: Primary register pairs except AF

1B .ERRX

Function: Displays an error message.
Input register: ACC: = Error code (the same as the error number listed in the error message table of the OWNER'S MANUAL). When the value of the 7-th bit of the accumulator is 1, the related device name is also displayed.
Output registers: None
Registers saved: Primary register pairs only

2C .DEVNM

Function: Interprets(specifies) the device name
Input registers: DE: = Pointer position of the character string which indicates the device name.
B: = The length of the string
Output registers: HL: = Pointer position next to the end of the device name which has been interpreted.
DE: = The starting address of the device table
ACC: = Device identification number (unit number)
Registers saved: None

2D .DEVFN

Function: Interprets the device name and file name.
Input register: DE: = Pointer position in the device name and file name string.
Output registers: None
Registers saved: None

2E .LUCHK

Function: Checks whether the logical number is defined.
Input registers: ACC: = Logical number
Output registers: ACC: = 1 (read opened)
ACC: = 2 (write opened)
ACC: = 3 (read/write opened)
CF: = 1 (not opened)
Registers saved: Primary registers except AF

2F .LOPEN

Function: Opens files which are not divided into blocks as object files. To execute this function, the device name and filename must be specified with function .DEVFN in advance.
Input registers: None
Output registers: None
Registers saved: None

30 .LOADF

Function: Loads files which are not divided into blocks such as object files. To execute this function, the file must have been opened with functions .DEVFN and .LOPEN in advance.

Input registers: HL: =Loading address

Output registers: None

Registers saved: None

31 .SAVEF

Function: Saves files which are not divided into blocks such as object files. To execute this function, the file name must be specified with function .DEVFN.

Input registers: DE: =Starting address of the memory area to be saved
ELMD20(1014H): =File size in bytes
ELMD22(1916H): =Load address
ELMD24(1018H): =Execution address

Output registers: None

Registers saved: None

32 .VRFYF

Function: Compare the contents of the specified memory area with a file which are not divided into blocks such as object files. To execute this function, the file must be opened with .DEVFN and .LOPEN in advance.

Input registers: None

Output registers: None

Registers saved: None

33 .RWOPN

Function: Opens to read or write a file which are divided into blocks such as source files (files in ASCII code). To execute this function, the device name and file name must be specified with function .DEVFN in advance.
ZRWX(1043H): =1 for read-open
ZRWX: =2 for write-open

Input registers: None

Output registers: None

Registers saved: Primary register pairs except AF

35 .INMSG

Function: Inputs one line of data of the file opened which has been opened with function call .RWOPN.

Input registers: DE: =Starting address of the input buffer

Output register: B: =Input file size in bytes
CF: =1 when the file end (EOF) has been detected.

Registers saved: DE,HL

37 .PRSTR

Function: Writes the specified bytes of data (max. 255 characters) into the file which has been write opened with .RWOPN.

Input registers: DE: =Starting address of the data to be written.
B: =Data size in bytes

Output registers: None

Registers saved: Primary register pairs except AF

38 .CLKL

Function: Closes or kills the files opened.

Input registers: ACC: =Logical number of the file to be closed or killed
(When ACC: =0,all files opened are closed or killed.)
B: =0 for kill and B < > 0 for close

Output registers: None

Registers saved: Primary register pairs except AF

39 .DIR

Function: Displays or prints out the information concerning files stored on the disk or the contents of the directory. The device name must be specified with monitor call .DEVNM in advance.

Input registers: ACC: =0 Inputs the contents of directory into the directory buffer in the monitor.
ACC< >0 Outputs the directory in the directory buffer to the device specified with the value in the ACC.
ACC: =88H To the CRT
ACC: =89H To the printer
Otherwise, the directory is output to the file or device specified by the logical number set.

Output registers: None

Registers saved: Primary register pairs except AF

3A .SETDF

Function: Sets the default device.
Input registers: DE: = Starting address of the device table
ACC: = Device identification number (unit number)
These are output registers set by .DEVNM.
Output registers: None
Registers saved: Primary register pairs only

3C .FINIT

Function: Initializes the I/O handler routine in the monitor (this function is used by the INIT statement of BASIC). The device name must be specified with monitor call .DEVNM in advance.
Input registers: None
Output registers: None
Registers saved: None

43 .ERCVR

Function: Performs the error recovery operation and stops the motor of the MZ disk or floppy disk.
Input registers: None
Output registers: None
Registers saved: None

Examples of use of monitor calls

In following examples, it is assumed that the SVC macro has been defined.

MACRO SVC

RST 3

DEFB @1 → A function number is assigned to parameter @1.

ENDM

When you create programs using this monitor, please add the program below to those programs at the top of them.

```
LD HL,ERADR          ; Sets the address of the error handling routine of the program.
LD(ERRORP),HL
LD HL,hot-start       ; Sets the hot start address of the program.
LD HL,last            ; Sets the last address of the program.
LD(TEXTST),HL
LD(POOL),HL
LD(HL),0
INC(HL),
INC HL
LD(VARST),HL
LD(TMPEND),HL
LD DE,6000            ; Sets 600H for a floppy disk and 800H for the MZ disk.
ADD HL,DE
LD(MEMLMT),HL
LD SP,HL              ; Sets HL to the initial value of the stack pointer.
```

Creates error handling routine ERADR mentioned above as follows

```
ERADR:               OR A
                     JR Z,break-adr; Jumps to SHIFT + BREAK handling routine.
                     CP 80H
                     JR Z,break-adr
                     SVC .ERR;Displays an error message.
```

Loading or verifying an object file

```
LD DE,FILE,X          ; Set name of object file to be loaded in DE.
SVC .COUNT           ; Set length of file name in B and returns.
SVC .DEVFN            ; Interpret (specify) device name and file name.
SVC .LOPEN            ; Open the file.
LD A,(ELDM)           ; Set file mode of file opened in ACC.
CP 1                  ; Object file?
JP NZ,error           ; Jump to error handling routine if not object file.
LD HL,(ELMD22)         ; Set load address in HL
SVC .LOADF(or SVC .VRFYF) ; Load or verify the file.
...
FILE. X: DEFM "QD: SAMPLE"
        DEFB0          ; Set the end of the file name to 0.
```

Saving an object file

```
LD DE,FILE.X          ; Set the name of the file to be saved in DE.
SVC.COUNT             ; Set the length of the file name in B and returns.
SVC.DEVFN             ; Interpret (specify) the device name and file name.
LD A,1
LD(ELMD),A            ; Set the file mode to the object file mode.
LD HL,length
LD(ELMD20),HL         ; Set the file length in bytes.
LD HL>Loading-adr
LD(ELMD22),HL         ; Set the load address of the file.
LD HL,exer-adr
LD(ELMD24),HL         ; Set the execution address of the file.
LD DE,save-adr
SVC.SAVEF             ; Save the file.
...
FILE. X: DEFM "QD: SAMPLE"
        DEFB0          ; Add 0 to end of filename.
```

Opening a source file (ASCII file)

```
LD A,open-mode           ; Read-open if the open mode is 1 and write-opens if it is 2.
LD(ZRWX),A
LD A,3
LD(ELMD),A               ; Set the file mode to the source file mode.
LD A,1
LD(ZLOG),A               ; Set the logical number to 1.
LD DE,FILE.X             ; Set the name of the file to be opened in DE.
SVC .COUNT              ; Sets the file length in bytes in B.
SVC .DEVFN               ; Interpret (specify) the device name and file name.
SVC .RWOPN                ; Open the file
LD A,(ELMD)               ; Set the file mode opened in A.
CP 3                     ; Source file?
JP NZ,error              ; Jump to the error handling routine if not source file.
:
:
FILE.X: DEFM "QD: SAMPLE"
      DEFB0               ; Set the end of the file name to 0.
                        ; Add 0 to end of filename.
```

Inputting one line of source file(ASCII file)

```
:                           ; The source file is assumed to be read-opened.
LD DE,buffer-adr          ; Set the starting address of the input buffer in DE.
SVC .INMSG                ; Input one line.
JP C,eof                  ; Perform the file end processing if CF: =
:                           ; Set the length of the line read in bytes in B.
:                           ; (OD at the end of the line is not included.)
```

Output of a source file (ASCII file)

```
:                           ; The source file is assumed to have been opened.
LD DE,save-adr            ; Starting address of the memory area to be saved.
LD B,length               ; File size in bytes (including OD at the end of each line).
SVC .PRSTR
:
```

Closing source files (ASCII files)

```
LD A,logical-number       ; Specified file only if the logical number is not zero and all files if it is
                           ; zero.
LD B,FFH
SVC .CLKL                 ; Closes the file(s).
```

Killing source file

LD A,logical-number ; Specified file only if the logical number is not zero and all files if it is
; zero.
LD B,0
SVC .CLKL ; Kill the file(s).

Setting a default device

LD DE,device-name ; Set the pointer position in the default device name in DE.
?SVC .COUNT ; Sets the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
SVC .SETDF ; Set the device specified by the value in ACC to the current device.

Display or Print out of directory

LD DE,device name ; Set the pointer position of the device name whose directory is to be
; displayed or printed out in DE.
SVC .COUNT ; Sets the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
LD B,A
XOR A ; Read the directory into the directory buffer in the monitor.
SVC .DIR
LD A,B
LD A,88H(or 89H)
SVC .DIR ; Output the directory to the CRT if the contents of ACC is 88H or to the
; printer if it is 89H.

Initialization of device (used by INIT statement of BASIC)

LD DE, device-name ; Set the pointer position for the device name to be initialized in DE.
SVC .COUNT ; Set the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
SVC .FINIT ; Initialize the device.

3-1 Monitor <1Z-013B>

** Z80 ASSEMBLER Z7-004C <1Z-013B> PAGE 01

03.06.85

** Z80 ASSEMBLER Z7-004C <1Z-013B> PAGE 02

03.06.85

3 Assemble list

```

01 0000      ;
02 0000      ;
03 0000      ; MONITOR PROGRAM 1Z-013B
04 0000      ;
05 0000      ; (MZ-800) FOR PAL
06 0000      ;
07 0000      ; REV. 1984.07.03
08 0000      ;
09 0000      ;
10 0000      ;
11 0000 C300E8 MONIT: ENT
12 0003      JP      E800H      ; MONITOR ON
13 0003 C3E607 GETL: ENT
14 0006      JP      ?GETL      ; GET LINE (END' CR')
15 0006 C30E09 LETNL: ENT
16 0009      JP      ?LTNL      ; NEW LINE
17 0009 C31809 NL: ENT
18 000C      JP      ?NL        ;
19 000C C32009 PRNTS: ENT
20 000F      JP      ?PRTS      ; PRINT SPACE
21 000F C32409 PRNTT: ENT
22 0012      JP      ?PRTT      ; PRINT TAB
23 0012 C33509 PRNT: ENT
24 0015      JP      ?PRNT      ; 1 CHARACTER PRINT
25 0015 C39308 MSG: ENT
26 0018      JP      ?MSG       ; 1 LINE PRINT (END' 0DH')
27 0018 C3A108 MSGX: ENT
28 001B      JP      ?MSGX      ; RST 3
29 001B C3BD08 GETKY: ENT
30 001E      JP      ?GET       ; GET KEY
31 001E C3320A BRKEY: ENT
32 0021      JP      ?BRK       ; GET BREAK
33 0021 C33604 WRINF: ENT
34 0024      JP      ?WRI       ; WRITE INFORMATION
35 0024 C37504 WRDAT: ENT
36 0027      JP      ?WRD       ; WRITE DATA
37 0027 C3D804 RDINF: ENT
38 002A      JP      ?ROI       ; READ INFORMATION
39 002A C3F804 RDDAT: ENT
40 002D      JP      ?RDD       ; READ DATA
41 002D C38805 VERFY: ENT
42 0030      JP      ?VRFY      ; VERIFING CMT
43 0030 C3C701 MELDY: ENT
44 0033      JP      ?MLDY      ; RST. 6
45 0033 C30803 TIMST: ENT
46 0036      JP      ?TMST      ; TIME SET
47 0037      NOP
48 0038 C33810      NOP
49 003B      JP      1038H      ; INTERRUPT ROUTINE
50 003B C35803 TIMRD: ENT
51 003E      JP      ?TMRD      ; TIME READ
52 003E C37705 BELL: ENT
53 0041      JP      ?BEL       ; BELL ON
54 0041 C3E502 XTEMP: ENT
55 0044      JP      ?TEMP      ; TEMPO SET (17)
56 0044 C3AB02 MSTA: ENT
57 0047      JP      MLDST      ; MELODY START
58 0047 C3BE02 MSTP: ENT
59 004A      JP      MLDSP      ; MELODY STOP
60 004A      ;

```

```

01 004A      ;
02 004A      ; START: ENT
03 004A 31F010 LD      SP,SP      ; STACK SET (10F0H)
04 004D ED56 IM      1      ; IM 1 SET
05 004F CD3E07 CALL   ?MODE      ; 8255,8253 MODE SET
06 0052 CD320A CALL   ?BRK      ; CTRL ?
07 0055 3019 JR      NC,ST0
08 0057 FE20 CP      20H      ; KEY IS CTRL KEY
09 0059 2015 JR      NZ,ST0
10 005B      ;
11 005B D3E1 CMY0: ENT
12 005D 11F0FF OUT     (E1H),A      ; D000H-FFFFH IS DRAM
13 0060 216B00 LD      DE,FFFF0H      ; TRANS. ADR.
14 0063 010500 LD      HL,$MCP      ; MEMORY CHANG PROGRAM
15 0066 EDB0 LD      BC,05      ; BYTE SIZE
16 0068 C3F0FF LDIR    JP      FFF0H      ; JUMP $FFFF0
17 006B      ;
18 006B      ; $MCP: ENT
19 006B D3E0 DEFW    E0D3H      ; 0000H-0FFFH IS DRAM
20 006D C300 DEFW    00C3H      ; OUT (E0H),A
21 006F 00 DEFB     00H      ; JP 0000H
22 0070      ;
23 0070      ; ST0: ENT
24 0070 06FF LD      B,FFH      ; BUFFER CLEAR
25 0072 21F110 LD      HL,NAME      ; 10F1H-11F0H CLEAR
26 0075 CD080F CALL   ?CLR      ;
27 0078 3E16 LD      A,16H      ; LASTER CLR.
28 007A CD1200 CALL   PRNT      ;
29 007D 3E71 LD      A,71H      ; BACK:BLUE CHA.:WRITE
30 007F 2100D8 LD      HL,D800H      ; COLOR ADDRESS
31 0082 CD0509 CALL   #CLR8      ;
32 0085 218D03 LD      HL,TIMIN      ; INTERRUPT JUMP ROUTINE
33 0088 3EC3 LD      A,C3H      ;
34 008A 323810 LD      (1038H),A
35 008D 223910 LD      (1039H),HL
36 0090 3E04 LD      A,04      ; NORMAL TEMPO
37 0092 329E11 LD      (TEMPW),A
38 0095 CDBE02 CALL   MLDSP      ; MELODY STOP
39 0098 CD0900 CALL   NL
40 009B 11E706 LD      DE,MSG?3      ; ** MONITOR 1Z-013B **
41 009E DF RST      3      ; CALL MGX
42 009F CD7705 CALL   ?BEL
43 00A2      ;
44 00A2 3E01 SS: ENT
45 00A4 329D11 LD      A,01H
46 00A7 2100E8 LD      (SWRK),A      ; KEY IN SILENT
47 00AA 77 LD      HL,E800H      ; USR ROM ?
48 00AB 1855 LD      (HL),A      ; ROM CHECK
49 00AD      ;
50 00AD CD0900 ST1: ENT
51 00B0 3E2A CALL   NL
52 00B2 CD1200 LD      A,2AH      ; '*' PRINT
53 00B5 11A311 LD      PRNT
54 00B8 CD0300 LD      DE,BUFER      ; GET LINE WORK (11A3H)
55 00BB 1A CALL   GETL
56 00BC 13 LD      A,(DE)
57 00BD FE0D INC     DE
58 00BF 28EC CP      0DH
59 00C1 FE4A JR      Z,ST1
60 00C3 282E CP      'J'      ; JUMP

```

```

01 00C5 FE4C      CP      'L'
02 00C7 2848      JR      Z,LOAD
03 00C9 FE46      CP      'F'
04 00CB 2832      JR      Z,FD
05 00CD FE42      CP      'B'
06 00CF 2826      JR      Z,SG
07 00D1 FE23      CP      'H'
08 00D3 2886      JR      Z,CMY0
09 00D5 FE50      CP      'P'
10 00D7 287C      JR      Z,PTEST
11 00D9 FE4D      CP      'M'
12 00DB CAA807     JP      Z,MCOR
13 00DE FE53      CP      'S'
14 00E0 CA5E0F     JP      Z,SAVE
15 00E3 FE56      CP      'U'
16 00E5 CACB0F     JP      Z,VRFY
17 00E8 FE44      CP      'D'
18 00EA CA290D     JP      Z,DUMP
19 00ED           ;
20 00ED           ;
21 00E0           DEFS    +4
22 00F1           ;
23 00F1 18C8      JR      ST2
24 00F3           ;
25 00F3           JUMP COMMAND
26 00F3           ;
27 00F3 CD3001     GOTO:  CALL  HEXIY
28 00F6 E9        JP      (HL)
29 00F7           ;
30 00F7           ; KEY SOUND ON OFF
31 00F7           ;
32 00F7 3A9D11     SG:    LD      A,(SRK)
33 00FA 1F        RRA
34 00FB 3F        CCF
35 00FC 17        RLA
36 00FD 18A5      JR      SS+2
37 00FF           ;
38 00FF           ; FLOPPY
39 00FF           ;
40 00FF 2100F0     FD:    LD      HL,F000H
41 0102 7E        F02:   LD      A,(HL)
42 0103 B7        OR      A
43 0104 20A7      JR      NZ,ST1
44 0106 E9        F01:   JP      (HL)
45 0107           ;
46 0107           ;
47 0107           ; ERROR (LOADING)
48 0107           ;
49 0107           ;
50 0107 FE02      ?ER:   ENT     02H
51 0109 28A2      CP      Z,ST1
52 010B 114701    LD      DE,MSGE1
53 010E DF        RST     3
54 010F 189C      JR      ST1
55 0111           ;
56 0111           ;
57 0111           ; LOAD COMMAND
58 0111           ;
59 0111 CDD804     LOAD:   CALL  ?RDI
60 0114 38F1      JR      C,?ER

```

```

; LOAD PROGRAM
; FLOPPY ACCESS
; KEY IN BELL
; CHANG MEMORY
; PRINTER TEST
; MEMORY CORRECTION
; SAVED DATA
; VERIFYING DATA
; DUMP DATA
; NOT COMMAND
; DB = SOUND WORK
; CHENGE MODE
; FLOPPY I/O CHECK
; A=02H : BREAK IN
; CHECK SUM ERROR
; CALL MSGX

```

```

01 0116 CD0900     LOAD:  CALL  NL
02 0119 11A009     LD      DE,MSG?2
03 011C DF        RST     3
04 011D 11F110     LD      DE,NAME
05 0120 DF        RST     3
06 0121 CDF804     CALL  ?R00
07 0124 38E1      JR      C,?ER
08 0126 2A0611     LD      HL,(EXADR)
09 0129 7C        LD      A,H
10 012A FE12      CP      12H
11 012C 38E1      JR      C,LOAD-2
12 012E E9        JP      (HL)
13 012F           ;
14 012F           ;
15 012F           ;
16 012F           ; GETLINE AND BREAK IN CHECK
17 012F           ;
18 012F           ; EXIT BREAK IN THEN JUMP (ST1)
19 012F           ; ACC=TOP OF LINE DATA
20 012F           ;
21 012F           ;
22 012F E3        BGETL:  ENT     (SP),HL
23 0130 C1        EX      BC
24 0131 11A311     POP     DE,BUFER
25 0134 CD0300     CALL  GETL
26 0137 1A        LD      A,(DE)
27 0138 FE18      CP      18H
28 013A 28D3      JR      Z,LOAD-2
29 013C E9        JP      (HL)
30 013D           ;
31 013D           ; ASCII TO HEX CONVERT
32 013D           ; INPUT (DE)=ASCII
33 013D           ; CY=1 THEN JUMP (ST1)
34 013D           ;
35 013D           ;
36 013D F0E3      HEXIY:  ENT     (SP),IY
37 013F F1        EX      AF
38 0140 CD1004     POP     HLHEX
39 0143 38CA      CALL  C,LOAD-2
40 0145 F0E9      JR      (IY)
41 0147           ;
42 0147           ;
43 0147           ;
44 0147           ; MSGE1:  ENT
45 0147 43484543   DEFM    'CHECK SUM ER.'
46 0148 4B205355   ;
47 014F 4D204552   ;
48 0153 2E        DEFB    0DH
49 0154 00        ;
50 0155           ;
51 0155           ;
52 0155           ; PLOTTER PRINTER TEST COMMAND
53 0155           ; (OPG23)
54 0155           ;
55 0155           ; &=CONTROL COMMANDS GROUP
56 0155           ; C=PEN CHENGE
57 0155           ; G=GRAPH MODE
58 0155           ; S=80 CHA. IN 1 LINE
59 0155           ; L=40 CHA. IN 1 LINE
60 0155           ; T=PLOTTER TEST
; IN (DE)=PRINT DATA

```

```

01 0155      ;
02 0155      PTEST: ENT
03 0155 1A      LD A,(DE)
04 0156 FE26      CP '8'
05 0158 2016      JR NZ,PTST1
06 015A 13      PTST0: INC DE
07 015B 1A      LD A,(DE)
08 015C FE4C      CP 'L' ; 80 IN 1 LINE
09 015E 2816      JR Z,.LPT
10 0160 FE53      CP 'S' ; 80 IN 1LINE
11 0162 2817      JR Z,..LPT
12 0164 FE43      CP 'C' ; PEN CHENGE
13 0166 2823      JR Z,PEN
14 0168 FE47      CP 'G' ; GRAPH MODE
15 016A 2818      JR Z,PLOT
16 016C FE54      CP 'T' ; TEST
17 016E 2810      JR Z,PTRN
18 0170      ;
19 0170 CDA501      PTST1: CALL PMSG ; PLOT MESSAGE
20 0173 C3AD00      JP ST1
21 0176      ;
22 0176 117004      .LPT: LD DE,LLPT ; 01-09-09-0B-0D
23 0179 18F5      JR PTST1
24 017B      ;
25 017B 11D503      ..LPT: LD DE,SLPT ; 01-09-09-09-0D
26 017E 18F0      JR PTST1
27 0180      ;
28 0180 3E04      PTRN: LD A,04H ; TEST PATTERN
29 0182 1802      JR PLOT+2
30 0184      ;
31 0184 3E02      PLOT: LD A,02H ; GRAPH CODE
32 0186 CD8F01      CALL LPRNT
33 0189 18CF      JR PTST0
34 018B      ;
35 018B 3E1D      PEN: LD A,1DH ; 1 CHENGE CODE (TEXT MO
DE)      JR PLOT+2
36 018D 18F7      ;
37 018F      ;
38 018F      ; 1CHA. PRINT TO $LPT
39 018F      ;
40 018F      ; IN: ACC PRINT DATA
41 018F      ;
42 018F      ;
43 018F      ;
44 018F 0E00      LPRNT: LD C,0 ; RDA TEST
45 0191 47      LD B,A ; PRINT DATA STORE
46 0192 CDB601      CALL RDA
47 0195 78      LD A,B
48 0196 D3FF      OUT (FFH),A ; DATA OUT
49 0198 3E80      LD A,80H ; RDP HIGH
50 019A D3FE      OUT (FEH),A
51 019C 0E01      LD C,01H ; RDA TEST
52 019E CDB601      CALL RDA
53 01A1 AF      XOR A ; RDP LOW
54 01A2 D3FE      OUT (FEH),A
55 01A4 C9      RET
56 01A5      ;
57 01A5      ; $LPT MSG.
58 01A5      ; IN: DE DATA LOW ADR.
59 01A5      ; 0DH MSG. END
60 01A5      ;

```

```

01 01A5 D5      PMSG: PUSH DE
02 01A6 C5      PUSH BC
03 01A7 F5      PUSH AF
04 01A8 1A      PMSG1: LD A,(DE) ; ACC=DATA
05 01A9 CD8F01      CALL LPRNT
06 01AC 1A      LD A,(DE)
07 01AD 13      INC DE
08 01AE FE0D      CP 0DH ; END ?
09 01B0 20F6      JR NZ,PMSG1
10 01B2 F1      POP AF
11 01B3 C1      POP BC
12 01B4 D1      POP DE
13 01B5 C9      RET
14 01B6      ;
15 01B6      ; RDA CHECK
16 01B6      ;
17 01B6      ; BRKEY IN TO MONITOR RETURN
18 01B6      ; IN: C RDA CODE
19 01B6      ;
20 01B6 DBFE      RDA: IN A,(FEH)
21 01B8 E60D      AND 0DH
22 01BA B9      CP C
23 01BB C8      RET Z
24 01BC CD1E00      CALL BRKEY
25 01BF 20F5      JR NZ,RDA
26 01C1 31F010      LD SP,SP
27 01C4 C3AD00      JP ST1
28 01C7      ;
29 01C7      ; ORG 01C7H
30 01C7      ;
31 01C7      ; MELODY
32 01C7      ;
33 01C7      ; DE=DATA LOW ADR.
34 01C7      ; EXIT CF=1 BREAK
35 01C7      ; CF=0 OK
36 01C7      ;
37 01C7      ;
38 01C7      ;
39 01C7 C5      ?MLDY: ENT
40 01C8 D5      PUSH BC
41 01C9 E5      PUSH DE
42 01CA 3E02      PUSH HL
43 01CC 32A011      LD A,02H
44 01CF 0601      LD (OCTV),A
45 01D1 1A      LD B,01
46 01D2 FE0D      MLD1: LD A,(DE) ; CR
47 01D4 283B      CP 0DH
48 01D6 FEC8      JR Z,MLD4 ; END MARK
49 01D8 2837      CP C8H
50 01DA FECF      JR Z,MLD4 ; UNDER OCTAVE
51 01DC 2827      CP CFH
52 01DE FE2D      JR Z,MLD2 ; '-'
53 01E0 2823      CP 2DH
54 01E2 FE2B      JR Z,MLD2 ; '+'
55 01E4 2827      CP 2BH
56 01E6 FED7      JR Z,MLD3 ; UPPER OCTAVE
57 01E8 2823      CP 07H
58 01EA FE23      JR Z,MLD3 ; "H" HANON
59 01EC 216C02      LD HL,MTBL
60 01EF 2004      JR NZ,+6

```

```

01 01F1 218402      LD      HL,M#TBL
02 01F4 13          INC      DE
03 01F5 CD1C02      CALL    ONPU      ; ONTYO SET
04 01F8 38D7        JR       C,MLD1
05 01FA CDC802      CALL    RYTHM
06 01FD 3815        JR       C,MLD5
07 01FF CDA802      CALL    MLDST    ; MELODY START
08 0202 41          LD       B,C
09 0203 18CC        JR       MLD1
10 0205 3E03        MLD2:    LD       A,+3
11 0207 32A011      LD       (OCTV),A
12 020A 13          INC      DE
13 020B 18C4        JR       MLD1
14 020D 3E01        MLD3:    LD       A,1
15 020F 18F6        JR       MLD2+2
16 0211 CDC802      MLD4:    CALL    RYTHM
17 0214 F5          MLD5:    PUSH    AF
18 0215 CDBE02      CALL    MLDSP
19 0218 F1          POP      AF
20 0219 C39B06      JP       RET3
21 021C
22 021C            ;      ONPU TO RATIO CONV
23 021C
24 021C            ;      EXIT (RATIO)=RATIO VALUE
25 021C            ;      C=ONTYO*TEMPO
26 021C
27 021C            ;
28 021C C5          ONPU:    ENT      BC
29 021D 0608        PUSH    BC
30 021F 1A          LD       B,8
31 0220 BE          ONP1:    LD       A,(DE)
32 0221 2809        CP       (HL)
33 0223 23          JR       Z,ONP2
34 0224 23          INC      HL
35 0225 23          INC      HL
36 0226 10F8        INC      HL
37 0228 37          DJNZ     -6
38 0229 13          SCF
39 022A C1          INC      DE
40 022B C9          POP      BC
41 022C 23          RET
42 022D 05          ONP2:    INC      HL
43 022E 5E          PUSH    DE
44 022F 23          LD       E,(HL)
45 0230 56          INC      HL
46 0231 EB          LD       D,(HL)
47 0232 7C          EX       DE,HL
48 0233 B7          LD       A,H
49 0234 2809        OR       A
50 0236 3AA011      JR       Z,+11
51 0239 30          LD       A,(OCTV)    ; 11A0H OCTAVE WORK
52 023A 2803        DEC      A
53 023C 29          JR       Z,+5
54 023D 18FA        ADD     HL,HL
55 023F 22A111      JR       -4
56 0242 21A011      LD       (RATIO),HL    ; 11A1H ONPU RATIO
57 0245 3602        LD       HL,OCTV
58 0247 2B          LD       (HL),2
59 0248 D1          DEC      HL
60 0249 13          POP      DE
                INC      DE

```

```

01 024A 1A          LD       A,(DE)
02 024B 47          LD       B,A
03 024C E6F0        AND      F0H    ; ONTYO ?
04 024E FE30        CP       30H
05 0250 2803        JR       Z,+5
06 0252 7E          LD       A,(HL)    ; HL=ONTYO
07 0253 1805        JR       +7
08 0255 13          INC      DE
09 0256 78          LD       A,B
10 0257 E60F        AND      0FH
11 0259 77          LD       (HL),A    ; HL=ONTYO
12 025A 219C02      LD       HL,OPTBL
13 025D 85          ADD     A,L
14 025E 6F          LD       L,A
15 025F 4E          LD       C,(HL)
16 0260 3A9E11      LD       A,(TEMPW)
17 0263 47          LD       B,A
18 0264 AF          XOR      A
19 0265 81          ONP3:    ADD     A,C
20 0266 10FD        DJNZ     -1
21 0268 C1          POP      BC
22 0269 4F          LD       C,A
23 026A AF          XOR      A
24 026B C9          RET
25 026C
26 026C            ;
27 026C            ;
28 026C 43          MTBL:    ENT      43H    ; C
29 026D 4608        DEFB     0846H
30 026F 44          DEFB     44H    ; D
31 0270 5F07        DEFW     075FH
32 0272 45          DEFB     45H    ; E
33 0273 9106        DEFW     0691H
34 0275 46          DEFB     46H    ; F
35 0276 3306        DEFW     0633H
36 0278 47          DEFB     47H    ; G
37 0279 8605        DEFW     0586H
38 027B 41          DEFB     41H    ; A
39 027C EC04        DEFW     04ECH
40 027E 42          DEFB     42H    ; B
41 027F 6404        DEFW     0464H
42 0281 52          DEFB     52H    ; R
43 0282 0000        DEFW     0
44 0284            M#TBL:    ENT
45 0284 43          DEFB     43H    ; #C
46 0285 CF07        DEFW     07CFH
47 0287 44          DEFB     44H    ; #D
48 0288 F506        DEFW     06F5H
49 028A 45          DEFB     45H    ; #E
50 028B 3306        DEFW     0633H
51 028D 46          DEFB     46H    ; #F
52 028E DA05        DEFW     05DAH
53 0290 47          DEFB     47H    ; #G
54 0291 3705        DEFW     0537H
55 0293 41          DEFB     41H    ; #A
56 0294 A504        DEFW     04A5H
57 0296 42          DEFB     42H    ; #B
58 0297 2304        DEFW     0423H
59 0299 52          DEFB     52H    ; #R
60 029A

```

```

01 029A 0000      DEFW 0
02 029C          OPTBL: ENT
03 029C 01       DEFB 1
04 029D 02       DEFB 2
05 029E 03       DEFB 3
06 029F 04       DEFB 4
07 02A0 06       DEFB 6
08 02A1 08       DEFB 8
09 02A2 0C       DEFB 0CH
10 02A3 10       DEFB 10H
11 02A4 18       DEFB 18H
12 02A5 20       DEFB 20H
13 02A6          ;
14 02A6          ;
15 02A6          ;
16 02A6          ;
17 02A6          ;
18 02A6          ;
19 02A6 13       .4DE: ENT
20 02A7 13       INC DE
21 02A8 13       INC DE
22 02A9 13       INC DE
23 02AA C9       RET
24 02AB          ;
25 02AB          ;
26 02AB          ;
27 02AB          ;
28 02AB          ;
29 02AB          ;
30 02AB          ;
31 02AB          ;
32 02AB 2AA111   MLDST: ENT
33 02AE 7C       LD HL,(RATIO)
34 02AF B7       LD A,H
35 02B0 280C     OR A
36 02B2 05       JR Z,MLDSP
37 02B3 EB       PUSH DE
38 02B4 2104E0   EX DE,HL
39 02B7 73       LD HL,CONT0
40 02B8 72       LD (HL),E
41 02B9 3E01     LD (HL),D
42 02BB D1       LD A,1
43 02BC 1806     POP DE
44 02BE          JR MLDS1
45 02BE          ;
46 02BE 3E36     MLDSP: ENT
47 02C0 3207E0   LD A,36H
48 02C3 AF       LD (CONTF),A
49 02C4 3208E0   XOR A
50 02C7 C9       LD (SUNDG),A
51 02C8          RET
52 02C8          ;
53 02C8          ;
54 02C8          ;
55 02C8          ;
56 02C8          ;
57 02C8          ;
58 02C8          ;
59 02C8          ;
60 02C8 2100E0   RYTHM: ENT
                    LD HL,KEYPA

```

INCREMENT DE REG.

ORG 02ABH ; MLDST

MELODY START & STOP

MLDST: ENT
LD HL,(RATIO)
LD A,H
OR A
JR Z,MLDSP
PUSH DE
EX DE,HL
LD HL,CONT0
LD (HL),E
LD (HL),D
LD A,1
POP DE
JR MLDS1

MLDSP: ENT
LD A,36H
LD (CONTF),A
XOR A
LD (SUNDG),A
RET

RYTHM

B=COUNT DATA

IN
EXIT CF=1 BREAK
CF=0 OK

RYTHM: ENT
LD HL,KEYPA

; MODE SET (8253 C0)
; E007H
; E008H
; TEHRO RESET

; E000H

```

01 02CB 36F8     LD (HL),F8H
02 02CD 23       INC HL
03 02CE 7E       LD A,(HL)
04 02CF E681     AND 81H
05 02D1 2002     JR NZ,+4
06 02D3 37       SCF
07 02D4 C9       RET
08 02D5 3A08E0   LD A,(TEMP)
09 02D8 0F       RRCA
10 02D9 38FA     JR C,-4
11 02DB 3A08E0   LD A,(TEMP)
12 02DE 0F       RRCA
13 02DF 30FA     JR NC,-4
14 02E1 10F2     DJNZ -12
15 02E3 AF       XOR A
16 02E4 C9       RET
17 02E5          ;
18 02E5          ;
19 02E5          ;
20 02E5          ;
21 02E5          ;
22 02E5          ;
23 02E5          ;
24 02E5 F5       ?TEMP: ENT
25 02E6 C5       PUSH AF
26 02E7 E60F     PUSH BC
27 02E9 47       AND 0FH
28 02EA 3E08     LD B,A
29 02EC 90       LD A,8
30 02ED 329E11   SUB B
31 02F0 C1       LD (TEMPW),A
32 02F1 F1       POP BC
33 02F2 C9       POP AF
34 02F3          RET
35 02F3          ;
36 02F3          ;
37 02F3          ;
38 02F3          ;
39 02F3          ;
40 02F3          ;
41 02F3          ;
42 02F3          ;
43 02F3 217311   CRT MANAGMENT
44 02F6 3A7211   EXIT HL: DSPXY H=Y,L=X
45 02F9 85       DE:MANG ADR. (ON DSPXY)
46 02FA 6F       A :MANG DATA
47 02FB 7E       CY:MANG=1
48 02FC 23       ;
49 02FD CB16     ;
50 02FF B6       ;
51 0300 CB1E     ;
52 0302 0F       ;
53 0303 EB       ;
54 0304 2A7111   ;
55 0307 C9       ;
56 0308          ;
57 0308          ;
58 0308          ;
59 0308          ;
60 0308          ;

```

; BREAK IN CHECK

; E008H
; TEMPO OUT

TEMPO SET

ACC=VALUE (1-7)

?TEMP: ENT
PUSH AF
PUSH BC
AND 0FH
LD B,A
LD A,8
SUB B
LD (TEMPW),A
POP BC
POP AF
RET

CRT MANAGMENT

EXIT HL: DSPXY H=Y,L=X
DE:MANG ADR. (ON DSPXY)
A :MANG DATA
CY:MANG=1

.MANG: ENT
LD HL,MANG
LD A,(1172H)
ADD A,L
LD L,A
LD A,(HL)
INC HL
RL (HL)
OR (HL)
RR (HL)
RRCA
EX DE,HL
LD HL,(DSPXY)
RET

ORG 0308H

; CRT MANG. POINTER
; DSPXY+1

```

01 0308      ; TIME SET
02 0308      ;
03 0308      ; ACC=0 : AM
04 0308      ; =1 : PM
05 0308      ; DE=SEC: BINARY
06 0308      ;
07 0308      ?TMS1: ENT
08 0308 F3    DI
09 0309 C5    PUSH BC
10 030A D5    PUSH DE
11 030B E5    PUSH HL
12 030C 329B11 LD (AMPM),A ; AMPM DATA
13 030F 3EF0   LD A,F0H
14 0311 329C11 LD (TIMEFG),A ; TIME FLAG
15 0314 21C0A8 LD HL,A8C0H ; 12H
16 0317 AF     XOR A
17 0318 ED52   SBC HL,DE ; COUNT DATA = 12H-IN DA
18 031A E5     PUSH HL
19 031B 00     NOP
20 031C EB     EX DE,HL
21 031D 2107E0 LD HL,CONTF ; E007H
22 0320 3674   LD (HL),74H
23 0322 36B0   LD (HL),B0H
24 0324 2B     DEC HL ; CONT2
25 0325 73     LD (HL),E
26 0326 72     LD (HL),D
27 0327 2B     DEC HL ; CONT1
28 0328 360A   LD (HL),0AH
29 032A 3600   LD (HL),0
30 032C 23     INC HL
31 032D 23     INC HL ; CONTF
32 032E 3680   LD (HL),80H
33 0330 2B     DEC HL ; CONT2
34 0331 4E     LD C,(HL)
35 0332 7E     LD A,(HL)
36 0333 BA     CP D
37 0334 20FB   JR NZ,?TMS1
38 0336 79     LD A,C
39 0337 B8     CP E
40 0338 20F7   JR NZ,?TMS1
41 033A 2B     DEC HL
42 033B 00     NOP
43 033C 00     NOP
44 033D 00     NOP
45 033E 36FB   LD (HL),FBH ; 1SEC
46 0340 363C   LD (HL),3CH
47 0342 23     INC HL
48 0343 D1     POP DE
49 0344 4E     LD C,(HL)
50 0345 7E     LD A,(HL)
51 0346 BA     CP D
52 0347 20FB   JR NZ,?TMS2
53 0349 79     LD A,C
54 034A B8     CP E
55 034B 20F7   JR NZ,?TMS2
56 034D E1     POP HL
57 034E D1     POP DE
58 034F C1     POP BC
59 0350 FB     EI
60 0351 C9     RET

```

```

01 0352      ;
02 0352      ;
03 0352      ; BELL DATA
04 0352      ;
05 0352      ; ?BELD: ENT
06 0352 D7    DEFB 07H
07 0353 4130   DEFM 'A0'
08 0355 00     DEFB 0DH
09 0356      ;
10 0356      ;
11 0356      ;
12 0356      ; DEFS +2
13 0358      ; ORG 0358H
14 0358      ;
15 0358      ; TIME READ
16 0358      ;
17 0358      ; EXIT ACC=0 :AM
18 0358      ; =1 :PM
19 0358      ; DE=SEC. BINARY
20 0358      ;
21 0358      ;
22 0358 E5     ?TMRD: ENT
23 0359 2107E0 PUSH HL
24 035C 3680   LD HL,CONTF
25 035E 2B     LD (HL),80H
26 035F F3     DEC HL ; CONT2
27 0360 5E     DI
28 0361 56     LD E,(HL)
29 0362 FB     LD D,(HL)
30 0363 7B     EI
31 0364 B2     LD A,E
32 0365 280E   OR D
33 0367 AF     JR Z,?TMR1
34 0368 21C0A8 XOR A
35 036B ED52   LD HL,A8C0H
36 036D 3810   SBC HL,DE
37 036F EB     JR C,?TMR2
38 0370 3A9B11 EX DE,HL
39 0373 E1     LD A,(AMPM)
40 0374 C9     POP HL
41 0375 11C0A8 RET
42 0378 3A9B11 LD DE,A8C0H ; 12H
43 037B EE01   LD A,(AMPM)
44 037D E1     XOR 1
45 037E C9     POP HL
46 037F F3     RET
47 0380 2106E0 ?TMR1: DI
48 0383 7E     LD HL,CONTF
49 0384 2F     LD A,(HL)
50 0385 5F     CPL
51 0386 7E     LD E,A
52 0387 2F     LD A,(HL)
53 0388 57     CPL
54 0389 FB     LD D,A
55 038A 13     EI
56 038B 18EB   INC DE
57 038D      JR ?TMR1+3
58 038D      ;
59 038D      ; TIME INTERRUPT
60 038D      ;
    TIMIN: ENT

```

```

01 038D F5          PUSH AF
02 038E C5          PUSH BC
03 038F D5          PUSH DE
04 0390 E5          PUSH HL
05 0391 219B11      LD HL,AMPM
06 0394 7E          LD A,(HL)
07 0395 EE01        XOR 1
08 0397 77          LD (HL),A
09 0398 2107E0      LD HL,CONTF
10 039B 3680        LD (HL),80H
11 039D 2B          DEC HL
12 039E E5          PUSH HL
13 039F 5E          LD E,(HL)
14 03A0 56          LD D,(HL)
15 03A1 21C0A8      LD HL,A8C0H
16 03A4 19          ADD HL,DE
17 03A5 2B          DEC HL
18 03A6 2B          DEC HL
19 03A7 EB          EX DE,HL
20 03A8 E1          POP HL
21 03A9 73          LD (HL),E
22 03AA 72          LD (HL),D
23 03AB E1          POP HL
24 03AC D1          POP DE
25 03AD C1          POP BC
26 03AE F1          POP AF
27 03AF FB          EI
28 03B0 C9          RET
29 03B1            ;
30 03B1            ; SPACE PRINT AND DISP ACC
31 03B1            ;
32 03B1            ; INPUT:HL=DISP. ADR.
33 03B1            ;
34 03B1            SPHEX: ENT
35 03B1 CD2009      CALL ?PRTS ; SP.PRINT
36 03B4 7E          LD A,(HL)
37 03B5 CDC303      CALL PRTHX ; DSP OF ACC (ASCII)
38 03B8 7E          LD A,(HL)
39 03B9 C9          RET
40 03BA            ;
41 03BA            ;
42 03BA            ;
43 03BA            ; ORG 03BAH
44 03BA            ;
45 03BA            ; (ASCII PRINT) FOR HL
46 03BA            ;
47 03BA            PRTHL: ENT
48 03BA 7C          LD A,H
49 03BB CDC303      CALL PRTHX
50 03BE 7D          LD A,L
51 03BF 1802        JR PRTHX
52 03C1            ;
53 03C1            ; DEFS +2
54 03C3            ; ORG 03C3H;PRTHX
55 03C3            ;
56 03C3            ; (ASCII PRINT) FOR ACC
57 03C3            ;
58 03C3            PRTHX: ENT
59 03C3 F5          PUSH AF
60 03C4 0F          RRCA

```

```

01 03C5 0F          RRCA
02 03C6 0F          RRCA
03 03C7 0F          RRCA
04 03C8 CDDA03      CALL ASC
05 03CB CD1200      CALL PRNT
06 03CE F1          POP AF
07 03CF CDDA03      CALL ASC
08 03D2 C31200      JP PRNT
09 03D5            ;
10 03D5            ;
11 03D5            ;
12 03D5            ;
13 03D5            ; 80 CHA. 1 LINE CODE (DATA)
14 03D5            ;
15 03D5            ; SLPT: ENT
16 03D5 01          DEFB 01H ; TEXT MODE
17 03D6 09          DEFB 09H
18 03D7 09          DEFB 09H
19 03D8 09          DEFB 09H
20 03D9 0D          DEFB 0DH
21 03DA            ;
22 03DA            ; ORG 03DAH;ASC
23 03DA            ;
24 03DA            ; HEXADECIMAL TO ASCII
25 03DA            ; IN : ACC (03-D0)=HEXADECIMAL
26 03DA            ; EXIT: ACC = ASCII
27 03DA            ;
28 03DA            ;
29 03DA E60F        AND 0FH
30 03DC FE0A        CP 0AH
31 03DE 3802        JR C,NOADD
32 03E0 C607        ADD A,7
33 03E2            NOADD: ENT
34 03E2 C630        ADD A,30H
35 03E4 C9          RET
36 03E5            ;
37 03E5            ; ASCII TO HEXADECIMAL
38 03E5            ; IN : ACC = ASCII
39 03E5            ; EXIT : ACC = HEXADECIMAL
40 03E5            ; CY = 1 ERROR
41 03E5            ;
42 03E5            ;
43 03E5            ; HEXJ: ENT
44 03E7 D8          SUB 30H
45 03E8 FE0A        RET C
46 03EA 3F          CP 0AH
47 03EB D0          CCF
48 03EC D607        RET NC
49 03EE FE10        SUB 7
50 03F0 3F          CP 10H
51 03F1 D8          CCF
52 03F2 FE0A        RET C
53 03F4 C9          CP 0AH
54 03F5            RET
55 03F5            ;
56 03F5            ;
57 03F9            ; ORG 03F9H;HEX
58 03F9            ;
59 03F9 18EA        ENT
60 03FB            JR HEXJ

```

```

01 03FB      ; PRASS PLAY MESSAGE
02 03FB      ;
03 03FB      MSG#1: ENT
04 03FB 7F20      DEFW 207FH
05 03FD      MSG#2: ENT
06 03FD 504C4159      DEFM 'PLAY'
07 0401 00      DEFB 0DH
08 0402      MSG#3: ENT
09 0402 7F20      DEFW 207FH
10 0404 5245434F      DEFM 'RECORD.'
11 0408 52442E      DEFB 0DH
12 040B 0D
13 040C      ;
14 040C      ;
15 040C      DEFS +4
16 0410      ;ORG 0410H;HLHEX
17 0410      ;
18 0410      ;
19 0410      ; 4 ASCII TO (HL)
20 0410      ;
21 0410      ; IN DE=DATA LOW ADR.
22 0410      ; EXIT CF=0 : OK
23 0410      ; =1 : OUT
24 0410      ;
25 0410      HLHEX: ENT
26 0410 D5      PUSH DE
27 0411 CD1F04      CALL 2HEX
28 0414 3807      JR C,+9
29 0416 67      LD H,A
30 0417 CD1F04      CALL 2HEX
31 041A 3801      JR C,+3
32 041C 6F      LD L,A
33 041D D1      POP DE
34 041E C9      RET
35 041F      ;
36 041F      ;ORG 041FH;2HEX
37 041F      ;
38 041F      ;
39 041F      ; 2 ASCII TO (ACC)
40 041F      ;
41 041F      ; IN DE=DATA LOW ADR.
42 041F      ;
43 041F      ; EXIT CF=0 : OK
44 041F      ; =1 : OUT
45 041F      ;
46 041F      2HEX: ENT
47 041F C5      PUSH BC
48 0420 1A      LD A,(DE)
49 0421 13      INC DE
50 0422 CDF903      CALL HEX
51 0425 380D      JR C,+15
52 0427 0F      RRCA
53 0428 0F      RRCA
54 0429 0F      RRCA
55 042A 0F      RRCA
56 042B 4F      LD C,A
57 042C 1A      LD A,(DE)
58 042D 13      INC DE
59 042E CDF903      CALL HEX
60 0431 3801      JR C,+3

```

```

01 0433 B1      OR C
02 0434 C1      2HE1: POP BC
03 0435 C9      RET
04 0436      ;
05 0436      ;
06 0436      ; WRITE INFORMATION
07 0436      ;
08 0436      ?WRI: ENT
09 0436 F3      DI
10 0437 D5      PUSH DE
11 0438 C5      PUSH BC
12 0439 E5      PUSH HL
13 043A 1607      LD D,D7H
14 043C 1ECC      LD E,CCH
15 043E 21F010      LD HL,IBUFE
16 0441 018000      LD BC,80H
17 0444 CD1A07      CALL CKSUM
18 0447 CD9F06      CALL MOTOR
19 044A 3818      JR C,WRI3
20 044C 7B      LD A,E
21 044D FECC      CP CCH
22 044F 200D      JR NZ,WRI2
23 0451 CD0900      CALL NL
24 0454 D5      PUSH DE
25 0455 116704      LD DE,MSG#7
26 0458 DF      RST 3
27 0459 11F110      LD DE,NAME
28 045C DF      RST 3
29 045D D1      POP DE
30 045E CD7A07      CALL GAP
31 0461 CD8A04      CALL WTAPE
32 0464 C35405      JP RET2
33 0467      ;
34 0467      ;
35 0467 57524954      MSG#7: ENT
36 046B 494E4720      DEFM 'WRITING'
37 046F 0D      DEFB 0DH
38 0470      ;
39 0470      ;
40 0470      ;
41 0470      ; 40 CHA. IN 1 LINE CODE (DATA)
42 0470      ;
43 0470      ;
44 0470 01      LLPT: ENT
45 0471 09      DEFB 01H
46 0472 09      DEFB 09H
47 0473 0B      DEFB 09H
48 0474 0D      DEFB 0BH
49 0475      DEFB 0DH
50 0475      ;
51 0475      ;
52 0475      ;
53 0475      ; WRITE DATA
54 0475      ;
55 0475      ; EXIT CF=0 : OK
56 0475      ; =1 : BREAK
57 0475      ;
58 0475      ;
59 0475 F3      ?WRD: ENT
60 0476 D5      DI
      PUSH DE

```

```

01 0477 C5          PUSH BC
02 0478 E5          PUSH HL
03 0479 16D7        LD D,D7H
04 047B 1E53        LD E,53H
05 047D ED4B0211    LD BC,(SIZE)
06 0481 2A0411      LD HL,(DTADR)
07 0484 78          LD A,B
08 0485 B1          OR C
09 0486 284A        JR Z,RET1
10 0488 18BA        JR WRI1
11 048A            ;
12 048A            ;
13 048A            ; TAPE WRITE
14 048A            ;
15 048A            ; BC=BYTE SIZE
16 048A            ; HL=DATA LOW ADR.
17 048A            ;
18 048A            ; EXIT CF=0 : OK
19 048A            ; =1 : BREAK
20 048A            ;
21 048A D5          WTAPE: PUSH DE
22 048B C5          PUSH BC
23 048C E5          PUSH HL
24 048D 1602        LD D,2
25 048F 3EF8        LD A,F8H
26 0491 3200E0      LD (KEYPA),A
27 0494 7E          WTAPE1: LD A,(HL)
28 0495 CD6707      CALL WBYTE
29 0498 3A01E0      LD A,(KEYPB)
30 049B E681        AND 81H
31 049D C2A504      JP NZ,WTAPE2
32 04A0 3E02        LD A,02H
33 04A2 37          SCF
34 04A3 182D        JR WTAPE3
35 04A5 23          WTAPE2: INC HL
36 04A6 0B          DEC BC
37 04A7 78          LD A,B
38 04A8 B1          OR C
39 04A9 C29404      JP NZ,WTAPE1
40 04AC 2A9711      LD HL,(SUMDT)
41 04AF 7C          LD A,H
42 04B0 CD6707      CALL WBYTE
43 04B3 7D          LD A,L
44 04B4 CD6707      CALL WBYTE
45 04B7 CD1A0A      CALL LONG
46 04BA 15          DEC D
47 04BB C2C204      JP NZ,+7
48 04BE B7          OR A
49 04BF C3D204      JP WTAPE3
50 04C2 0600        LD B,0
51 04C4 CD010A      CALL SHORT
52 04C7 05          DEC B
53 04C8 C2C404      JP NZ,-4
54 04CB E1          POP HL
55 04CC C1          POP BC
56 04CD C5          PUSH BC
57 04CE E5          PUSH HL
58 04CF C39404      JP WTAPE1
59 04D2            WTAPE3:
60 04D2 E1          RET1: POP HL

```

```

01 04D3 C1          POP BC
02 04D4 D1          POP DE
03 04D5 C9          RET
04 04D6            ;
05 04D6            ;
06 04D6            ;
07 04D6            ;
08 04D6            ;
09 04D8            ORG 04D8H
10 04D8            ;
11 04D8            ;
12 04D8            ; READ INFORMATION (FROM $CMT)
13 04D8            ;
14 04D8            ; EXIT ACC=0 : OK CF=0
15 04D8            ; =1 : ER CF=1
16 04D8            ; =2 : BREAK CF=1
17 04D8            ;
18 04D8            ; ?RDI: ENT
19 04D8 F3          DI
20 04D9 D5          PUSH DE
21 04DA C5          PUSH BC
22 04DB E5          PUSH HL
23 04DC 16D2        LD D,D2H
24 04DE 1ECC        LD E,CCH
25 04E0 018000      LD BC,80H
26 04E3 21F010      LD HL,IBUFE
27 04E6            RD1: ENT
28 04E6 CD9F06      CALL MOTOR
29 04E9 DA7205      JP C,RTP6
30 04EC CD5B06      CALL TMARK
31 04EF DA7205      JP C,RTP6
32 04F2 CD0E05      CALL RTAPE
33 04F5 C35405      JP RTP4
34 04F8            ;
35 04F8            ;
36 04F8            ;
37 04F8            ; ORG 04F8H
38 04F8            ;
39 04F8            ;
40 04F8            ; READ DATA (FROM $CMT)
41 04F8            ;
42 04F8            ; EXIT SAME UP
43 04F8            ;
44 04F8            ; ?RDD: ENT
45 04F8 F3          DI
46 04F9 D5          PUSH DE
47 04FA C5          PUSH BC
48 04FB E5          PUSH HL
49 04FC 16D2        LD D,D2H
50 04FE 1E53        LD E,53H
51 0500 ED4B0211    LD BC,(SIZE)
52 0504 2A0411      LD HL,(DTADR)
53 0507 78          LD A,B
54 0508 B1          OR C
55 0509 CA5405      JP Z,RTP4
56 050C 1808        JR RD1
57 050E            ;
58 050E            ;
59 050E            ; READ TAPE
60 050E            ;

```

```

01 050E      ; IN BC=SIZE
02 050E      ; DE=LOAD ADR.
03 050E      ;
04 050E      ; EXIT ACC=0 : OK CF=0
05 050E      ; =1 : ER =1
06 050E      ; =2 : BREAK=1
07 050E      ;
08 050E      ;
09 050E 05    RTAPE: ENT
10 050F C5    PUSH DE
11 0510 E5    PUSH BC
12 0511 2602  PUSH HL
13 0513      LD H,2
14 0513 0101E0 RTP1: ENT
15 0516 1102E0 LD BC,KEYPB
16 0519      LD DE,CSTR
17 0519 CD0106 RTP2: ENT
18 051C 3854   CALL EDGE
19 051E CD4A0A JR C,RTP6
20 0521 1A    CALL DLY3
21 0522 E620   LD A,(DE)
22 0524 CA1905 AND 20H
23 0527 54    JP Z,RTP2
24 0528 210000 LD D,H
25 052B 229711 LD HL,0
26 052E E1    LD (SUMDT),HL
27 052F C1    POP HL
28 0530 C5    POP BC
29 0531 E5    PUSH BC
30 0532      PUSH HL
31 0532 CD2406 RTP3: ENT
32 0535 383B   CALL RBYTE
33 0537 77    JR C,RTP6
34 0538 23    LD (HL),A
35 0539 08    INC HL
36 053A 78    DEC BC
37 053B B1    LD A,B
38 053C 20F4   OR C
39 053E 2A9711 JR NZ,RTP3
40 0541 CD2406 LD HL,(SUMDT)
41 0544 382C   CALL RBYTE
42 0546 5F    JR C,RTP6
43 0547 CD2406 LD E,A
44 054A 3826   CALL RBYTE
45 054C BD    JR C,RTP6
46 054D 2016   CP L
47 054F 7B    JR NZ,RTP5
48 0550 BC    LD A,E
49 0551 2012   CP H
50 0553      JR NZ,RTP5
51 0553 AF    RTP8: ENT
52 0554      XOR A
53 0554      ENT
54 0554 E1    RTP4: ENT
55 0555 C1    RET2: ENT
56 0556 D1    POP HL
57 0557 CD0007 POP BC
58 055A F5    POP DE
59 055B 3A9C11 CALL MSTOP
60 055E FEF0   PUSH AF
                LD A,(TIMFG)
                CP F0H

```

```

01 0560 2001      JR NZ,+3
02 0562 FB       EI
03 0563 F1       POP AF
04 0564 C9       RET
05 0565          ;
06 0565          RTP5: ENT
07 0565 15       DEC D
08 0566 2806     JR Z,RTP7
09 0568 62       LD H,D
10 0569 CDE20F   CALL GAPCK
11 056C 18A5     JR RTP1
12 056E          RTP7: ENT
13 056E 3E01     LD A,1
14 0570 1802     JR RTP9
15 0572          RTP6: ENT
16 0572 3E02     LD A,2
17 0574          RTP9: ENT
18 0574 37       SCF
19 0575 1800     JR RTP4
20 0577          ;
21 0577          ; BELL
22 0577          ;
23 0577          ;
24 0577          ?BEL: ENT
25 0577 D5       PUSH DE
26 0578 115203   LD DE,?BELD
27 057B F7       RST 6
28 057C D1       POP DE
29 057D C9       RET
30 057E          ;
31 057E          ; FLASING AND KEYIN
32 057E          ; EXIT:ACC INPUT KEY DATA(DSP.CODE)
33 057E          ; H=F0H THEN NO KEYIN(Z FLG.)
34 057E          ;
35 057E          ;
36 057E C0FF09   FLKEY: ENT
37 0581 CDCA08   CALL ?FLAS
38 0584 FEF0     CALL ?KEY
39 0586 C9       CP F0H
40 0587          RET
41 0587          ;
42 0587          ;
43 0587          ;
44 0587          ;
45 0587          ;
46 0588          ; DEFS +1
47 0588          ; ORG 0588H
48 0588          ;
49 0588          ;
50 0588          ; VERIFY (FROM $CMT)
51 0588          ;
52 0588          ; EXIT ACC =0 : OK CF=0
53 0588          ; =1 : ER CF=1
54 0588          ; =2 : BREAK CF=1
55 0588          ;
56 0588          ;
57 0588 F3       ?VRFY: ENT
58 0589 D5       DI
59 058A C5       PUSH DE
60 058B E5       PUSH BC
                PUSH HL
                LD BC,(SIZE)

```

```

01 0590 2A0411      LD      HL,(DTADR)
02 0593 1602        LD      D,D2H
03 0595 1E53        LD      E,53H
04 0597 78          LD      A,B
05 0598 B1          OR      C
06 0599 28B9        JR      Z,RTP4
07 059B CD1A07      CALL    CKSUM
08 059E CD9F06      CALL    MOTOR
09 05A1 38CF        JR      C,RTP6
10 05A3 CD5B06      CALL    TMARK
11 05A6 38CA        JR      C,RTP6
12 05A8 CDAD05      CALL    TVRFY
13 05AB 18A7        JR      RTP4
14 05AD             ;
15 05AD             ;
16 05AD             ; DATA VERIFY
17 05AD             ;
18 05AD             ; BC=SIZE
19 05AD             ; HL=DATA LOW ADR
20 05AD             ; CSMOT=CHECK SUM
21 05AD             ; EXIT ACC=0 : OK CF=0
22 05AD             ; =1 : ER =1
23 05AD             ; =2 : BREAK=1
24 05AD             ;
25 05AD             ;
26 05AD             ; TVRFY: ENT
27 05AD 05          PUSH    DE
28 05AE C5          PUSH    BC
29 05AF E5          PUSH    HL
30 05B0 2602        LD      H,2
31 05B2             ;
32 05B2 0101E0      TVF1:  LD      BC,KEYPB
33 05B5 1102E0      LD      DE,CSTR
34 05B8             ;
35 05B8 CD0106      TVF2:  ENT
36 05BB DA7205      CALL    EDGE
37 05BE CD4A0A      JP      C,RTP6
38 05C1 1A          CALL    DLY3
39 05C2 E620        LD      A,(DE)
40 05C4 CAB805      AND      20H
41 05C7 54          JP      Z,TVF2
42 05C8 E1          LD      D,H
43 05C9 C1          POP      HL
44 05CA C5          POP      BC
45 05CB E5          PUSH    BC
46 05CC             ; TVF3: ENT
47 05CC CD2406      CALL    RBYTE
48 05CF 38A1        JR      C,RTP6
49 05D1 BE          CP      (HL)
50 05D2 209A        JR      NZ,RTP7
51 05D4 23          INC     HL
52 05D5 0B          DEC     BC
53 05D6 78          LD      A,B
54 05D7 B1          OR      C
55 05D8 20F2        JR      NZ,TVF3
56 05DA 2A9911      LD      HL,(CSMDT)
57 05DD CD2406      CALL    RBYTE
58 05E0 BC          CP      H
59 05E1 208B        JR      NZ,RTP7
60 05E3 CD2406      CALL    RBYTE

```

; 'R'
; 'S'

; TAPE MARK DETECT

; CALL DLY2*3

```

01 05E6 B0          CP      L
02 05E7 2085        JR      NZ,RTP7
03 05E9 15          DEC     D
04 05EA CA5305      JP      Z,RTP8
05 05ED 62          LD      H,D
06 05EE 18C2        JR      TVF1
07 05F0             ;
08 05F0             ; FLASHING DATA LOAD
09 05F0             ;
10 05F0             ; ?LOAD: ENT
11 05F0 F5          PUSH    AF
12 05F1 3A8E11      LD      A,(FLASH)
13 05F4 CDB10F      CALL    ?PONT
14 05F7 77          LD      (HL),A
15 05F8 F1          POP      AF
16 05F9 C9          RET
17 05FA             ;
18 05FA             ;
19 05FA             ; NEW LINE AND PRINT HL REG.(ASCII)
20 05FA             ;
21 05FA             ; NLPHL: ENT
22 05FA CD0900      CALL    NL
23 05FD CDBA03      CALL    PRTHL
24 0600 C9          RET
25 0601             ;
26 0601             ;
27 0601             ; ORG 0601H;EDGE
28 0601             ;
29 0601             ;
30 0601             ; EDGE (TAPE DATA EDGE DETECT)
31 0601             ;
32 0601             ; BC=KEYPB ($E001)
33 0601             ; DE=CSTR ($E002)
34 0601             ; EXIT CF=0 OK : CF=1 BREAK
35 0601             ;
36 0601             ; EDGE: ENT
37 0601 3EF8        LD      A,F8H
38 0603 3200E0      LD      (KEYPA),A
39 0606 00          NOP
40 0607             ; EDG1: ENT
41 0607 0A          LD      A,(BC)
42 0608 E681        AND      81H
43 060A 2002        JR      NZ,+4
44 060C 37          SCF
45 060D C9          RET
46 060E 1A          LD      A,(DE)
47 060F E620        AND      20H
48 0611 20F4        JR      NZ,EDG1
49 0613             ; EDG2: ENT
50 0613 0A          LD      A,(BC)
51 0614 E681        AND      81H
52 0616 2002        JR      NZ,+4
53 0618 37          SCF
54 0619 C9          RET
55 061A 1A          LD      A,(DE)
56 061B E620        AND      20H
57 061D 28F4        JR      Z,EDG2
58 061F C9          RET
59 0620             ;
60 0620             ;

```

```

01 0620      DEFS    +4
02 0624      ;ORG 0624H;RBYTE
03 0624      ;
04 0624      ;
05 0624      ;    1 BYTE READ
06 0624      ;
07 0624      ;    EXIT  SUMDT=STORE
08 0624      ;    CF=1 : BREAK
09 0624      ;
10 0624      ;    CF=0 : DATA=ACC
11 0624      ;
12 0624      RBYTE:  ENT
13 0624 C5      PUSH  BC
14 0625 D5      PUSH  DE
15 0626 E5      PUSH  HL
16 0627 210008  LD     HL,0800H
17 062A 0101E0  LD     BC,KEYPB
18 062D 1102E0  LD     DE,CSTR
19 0630      RBY1:  ENT
20 0630 CD0106  CALL   EDGE
21 0633 0A5406  JP     C,RBY3
22 0636 CD4A0A  CALL   DLY3
23 0639 1A      LD     A,(DE)
24 063A E620    AND    20H
25 063C CA4906  JP     Z,RBY2
26 063F E5      PUSH  HL
27 0640 2A9711  LD     HL,(SUMDT)
28 0643 23      INC    HL
29 0644 229711  LD     (SUMDT),HL
30 0647 E1      POP    HL
31 0648 37      SCF
32 0649      RBY2:  ENT
33 0649 7D      LD     A,L
34 064A 17      RLA
35 064B 6F      LD     L,A
36 064C 25      DEC    H
37 064D C23006  JP     NZ,RBY1
38 0650 CD0106  CALL   EDGE
39 0653 7D      LD     A,L
40 0654      RBY3:  ENT
41 0654 E1      POP    HL
42 0655 D1      POP    DE
43 0656 C1      POP    BC
44 0657 C9      RET
45 0658      ;
46 0658      ;
47 0658      ;    TAPE MARK DETECT
48 0658      ;
49 0658      ;    E=0L0 : INFORMATION
50 0658      ;    =0S0 : DATA
51 0658      ;    EXIT CF=0 :OK
52 0658      ;    =1 :BREAK
53 0658      ;
54 0658      ;
55 0658      DEFS    +3
56 065B      TMARK: ENT
57 065B      ;
58 065B      ;ORG 065BH
59 065B      ;
60 065B CDE20F  CALL   GAPCK

```

```

; KEY DATA $E001
; $TAPE DATA $E002
; 41 OR 101
; 13
; 20+18*63+33
; DATA READ :8

```

```

01 065E C5      PUSH  BC
02 065F D5      PUSH  DE
03 0660 E5      PUSH  HL
04 0661 212828  LD     HL,2828H
05 0664 7B      LD     A,E
06 0665 FECC      CP    CCH
07 0667 2803      JR    Z,+5
08 0669 211414  LD     HL,1414H
09 066C 229511  LD     (TMCNT),HL
10 066F 0101E0  LD     BC,KEYPB
11 0672 1102E0  LD     DE,CSTR
12 0675      TM1:  ENT
13 0675 2A9511  LD     HL,(TMCNT)
14 0678      TM2:  ENT
15 0678 CD0106  CALL   EDGE
16 067B 381E      JR    C,TM4
17 067D CD4A0A  CALL   DLY3
18 0680 1A      LD     A,(DE)
19 0681 E620      AND    20H
20 0683 28F0      JR    Z,TM1
21 0685 25      DEC    H
22 0686 20F0      JR    NZ,TM2
23 0688      TM3:  ENT
24 0688 CD0106  CALL   EDGE
25 068B 380E      JR    C,TM4
26 068D CD4A0A  CALL   DLY3
27 0690 1A      LD     A,(DE)
28 0691 E620      AND    20H
29 0693 20E0      JR    NZ,TM1
30 0695 2D      DEC    L
31 0696 20F0      JR    NZ,TM3
32 0698 CD0106  CALL   EDGE
33 069B      RET3:  ENT
34 069B      TM4:  ENT
35 069B E1      POP    HL
36 069C D1      POP    DE
37 069D C1      POP    BC
38 069E C9      RET
39 069F      ;
40 069F      ;
41 069F      ;    MOTOR ON
42 069F      ;
43 069F      ;    IN  D=0WA :WRITE
44 069F      ;    =0RA :READ
45 069F      ;    EXIT CF=0 :OK
46 069F      ;    =1 :BREAK
47 069F      ;
48 069F C5      MOTOR: ENT
49 06A0 D5      PUSH  BC
50 06A1 E5      PUSH  DE
51 06A2 060A      PUSH  HL
52 06A4      LD     B,10
53 06A4 3A02E0  LD     A,(CSTR)
54 06A7 E610      AND    10H
55 06A9 280E      JR    Z,MOT4
56 06AB      MOT1:  ENT
57 06AB 06FF      LD     B,FFH
58 06AD CD9609  CALL   DLY12
59 06B0 1802      JR    +4
60 06B2 18EB      JR    MOTOR

```

; ORG 065EH

; 'L'

; CALL DLY2*3

; CALL DLY2*3

; 2 SEC DELAY
; 7 MSEC DELAY
; MOTOR ENTRY ADJUST
; ORG 06B2H

79

```

01 070A 280B                                Z,MST3
02 070C                                ENT
MST2: LD A,06H
03 070C 3E06                                LD (CSTPT),A
04 070E 3203E0                            INC A
05 0711 3C                                LD (CSTPT),A
06 0712 3203E0                            DJNZ MST1
07 0715 10EE                                ENT
MST3: JP ?RSTR1
08 0717                                ;
09 0717 C3E60E                            ;
10 071A                                ;
11 071A                                ;
12 071A                                ;
13 071A                                ;
14 071A                                CHECK SUM
15 071A                                ;
16 071A                                IN BC=SIZE
17 071A                                HL=DATA ADR.
18 071A                                EXIT SUMDT=STORE
19 071A                                CSMDT=STORE
20 071A                                ;
21 071A                                CKSUM: ENT
22 071A C5                                PUSH BC
23 071B D5                                PUSH DE
24 071C E5                                PUSH HL
25 071D 110000                            LD DE,0
26 0720                                CKS1: ENT
27 0720 78                                LD A,B
28 0721 B1                                OR C
29 0722 200B                            JR NZ,CKS2
30 0724 EB                                EX DE,HL
31 0725 229711                          LD (SUMDT),HL
32 0728 229911                          LD (CSMDT),HL
33 072B E1                                POP HL
34 072C D1                                POP DE
35 072D C1                                POP BC
36 072E C9                                RET
37 072F                                CKS2: ENT
38 072F 7E                                LD A,(HL)
39 0730 C5                                PUSH BC
40 0731 0608                            LD B,+8
41 0733                                CKS3: ENT
42 0733 07                                RLCA
43 0734 3001                            JR NC,+3
44 0736 13                                INC DE
45 0737 10FA                            DJNZ CKS3
46 0739 C1                                POP BC
47 073A 23                                INC HL
48 073B 0B                                DEC BC
49 073C 18E2                            JR CKS1
50 073E                                ;
51 073E                                MODE SET OF KEYPORT
52 073E                                ;
53 073E                                ?MODE: ENT
54 073E 2103E0                          LD HL,KEYPF
55 0741 368A                            LD (HL),8AH ; 10001010
56 0743 3607                            LD (HL),07H ; PC3=1
57 0745 3605                            LD (HL),05H ; PC2=1
58 0747                                VGOFF: ENT
59 0747                                ;
60 0747 C9                                RET

```

```

01 0748 ;
02 0748 ;
03 0748 DEFS +17
04 0759 ;
05 0759 ;ORG 0759H;DLY1
06 0759 ;
07 0759 ; 107 MICRO SEC DELY
08 0759 ;
09 0759 DLY1: ENT
10 0759 3E1B LD A,1BH ; 111.39 micro sec
11 075B 3D DEC A
12 075C C25B07 JP NZ,-1
13 075F C9 RET
14 0760 ;
15 0760 ;ORG 0760H;DLY2
16 0760 ;
17 0760 DLY2: ENT
18 0760 3E19 LD A,19H ; 103.45 micro sec
19 0762 3D DEC A
20 0763 C26207 JP NZ,-1
21 0766 C9 RET
22 0767 ;
23 0767 ;
24 0767 ;
25 0767 ;
26 0767 ;
27 0767 ; 1 BYTE WRITE
28 0767 ;
29 0767 WBYTE: ENT
30 0767 C5 PUSH BC
31 0768 0608 LD B,+8
32 076A CD1A0A CALL LONG
33 076D WBY1: ENT
34 076D 07 RLCA
35 076E DC1A0A CALL C, LONG
36 0771 D4010A CALL NC, SHORT
37 0774 05 DEC B
38 0775 C26D07 JP NZ, WBY1
39 0778 C1 POP BC
40 0779 C9 RET
41 077A ;
42 077A ;
43 077A ; GAP + TAPEMARK
44 077A ;
45 077A ; E=0L0 LONG GAP
46 077A ; =0S0 SHORT GAP
47 077A ;
48 077A GAP: ENT
49 077A C5 PUSH BC
50 077B 05 PUSH DE
51 077C 7B LD A,E
52 077D 01F055 LD BC,55F0H
53 0780 112828 LD DE,2828H
54 0783 FECC CP CCH ; 'L'
55 0785 CA8E07 JP Z, GAP1
56 0788 01F82A LD BC,2AF8H
57 078B 111414 LD DE,1414H
58 078E GAP1: ENT
59 078E CD010A CALL SHORT
60 0791 0B DEC BC

```

```

01 0792 78 LD A,B
02 0793 B1 OR C
03 0794 20F8 JR NZ,-6
04 0796 GAP2: ENT
05 0796 CD1A0A CALL LONG
06 0799 15 DEC D
07 079A 20FA JR NZ,-4
08 079C GAP3: ENT
09 079C CD010A CALL SHORT
10 079F 1D DEC E
11 07A0 20FA JR NZ,-4
12 07A2 CD1A0A CALL LONG
13 07A5 D1 POP DE
14 07A6 C1 POP BC
15 07A7 C9 RET
16 07A8 ;
17 07A8 ; MEMORY CORRECTION
18 07A8 ; COMMAND 'M'
19 07A8 ;
20 07A8 MCR: ENT
21 07A8 CD3D01 CALL HEXIY ; CRRECTION ADR.
22 07AB MCR1: ENT
23 07AB CDFA05 CALL NPLHL ; COR. ADR. PRINT
24 07AE CDB103 CALL SPHEX ; ACC ASCII DISP.
25 07B1 CD2009 CALL ?PRTS ; SPACE PRINT
26 07B4 CD2F01 CALL BGETL ; GET DATA & CHECK DATA
27 07B7 CD1004 CALL HLHEX ; HLASCII(DE)
28 07BA 381B JR C,MCR3
29 07BC CDA602 CALL .4DE ; (INC DE)*4
30 07BF 13 INC DE
31 07C0 CD1F04 CALL 2HEX ; DATA CHECK
32 07C3 38E6 JR C,MCR1
33 07C5 BE CP (HL)
34 07C6 20E3 JR NZ,MCR1
35 07C8 13 INC DE
36 07C9 1A LD A,(DE)
37 07CA FE0D CP 0DH ; NOT CORRECTION ?
38 07CC 2806 JR Z,MCR2
39 07CE CD1F04 CALL 2HEX ; ACCHL(ASCII)
40 07D1 38D8 JR C,MCR1
41 07D3 77 LD (HL),A ; DATA CORRECT
42 07D4 MCR2: ENT
43 07D4 23 INC HL
44 07D5 18D4 JR MCR1
45 07D7 ;
46 07D7 60 MCR3: LD H,B ; MEMORY ADR.
47 07D8 69 LD L,C
48 07D9 18D0 JR MCR1
49 07DB ;
50 07DB ;
51 07DB ;
52 07DB ;
53 07DB ;
54 07E6 ORG 07E6H
55 07E6 ;
56 07E6 ;
57 07E6 ;
58 07E6 ; GET 1 LINE STATEMENT *
59 07E6 ; DE = DATA STORE LOW ADR.
60 07E6 ;

```

```

01 07E6          ;          (END =CR )
02 07E6          ;
03 07E6          ;
04 07E6          ?GETL: ENT
05 07E6 F5       PUSH    AF
06 07E7 C5       PUSH    BC
07 07E8 E5       PUSH    HL
08 07E9 D5       PUSH    DE
09 07EA          GETL1: ENT
10 07EA CDB309   CALL    ??KEY      ; ENTRY KEY
11 07ED          AUTO3: ENT
12 07ED F5       PUSH    AF        ; IN KEY DATA SAVE
13 07EE 47       LD      B,A
14 07EF 3A9D11   LD      A,(SWRK)   ; BELL WORK
15 07F2 0F       RRCA
16 07F3 D47705   CALL    NC,?BEL    ; ENTRY BELL
17 07F6 78       LD      A,B
18 07F7 217011   LD      HL,KANAF   ; KANA & GRAPH FLAG
19 07FA E6F0     AND     F0H
20 07FC FEC0     CP      C0H
21 07FE D1       POP     DE        ; Ereg=FLAGreg
22 07FF 78       LD      A,B
23 0800 2016     JR      NZ,GETL2
24 0802 FEC0     CP      CDH        ; CR
25 0804 2855     JR      Z,GETL3
26 0806 FECB     CP      CBH        ; BREAK
27 0808 CA2208   JP      Z,GETLC
28 080B FECF     CP      CFH        ; NIKO MARK WH.
29 080D 2809     JR      Z,GETL2
30 080F FEC7     CP      C7H        ; CRT EDITION
31 0811 300A     JR      NC,GETL5
32 0813 CB1B     RR      E          ; CY ?
33 0815 78       LD      A,B
34 0816 3005     JR      NC,GETL5
35 0818          GETL2: ENT
36 0818 CDB50D   CALL    ?DSP
37 081B 18CD     JR      GETL1
38 081D          GETL5: ENT
39 081D CDDC0D   CALL    ?DPCT      ; CRT CONTROL
40 0820 18C8     JR      GETL1
41 0822          ;
42 0822          ; BREAK IN
43 0822          ;
44 0822 E1       GETLC: POP     HL
45 0823 E5       PUSH    HL
46 0824 361B     LD      (HL),1BH   ; BREAK CODE
47 0826 23       INC     HL
48 0827 360D     LD      (HL),0DH
49 0829 1853     JR      GETLR
50 082B          ; GETLA
51 082B          ;
52 082B 0F       GETLA: RRCA      ; CYD7
53 082C 3037     JR      NC,GETL6
54 082E 1833     JR      GETLB
55 0830          ;
56 0830          ;
57 0830          ;
58 0830          ; DELAY 7M SEC AND SWEP
59 0830          ;
60 0830 CD9609   DSWEPT: CALL    DLY12

```

```

01 0833 CD500A   CALL    ?SWEP
02 0836 C9       RET
03 0837          ;
04 0837          ;
05 0837          ; DEFS    36
06 085B          ;
07 085B          ;
08 085B          ;
09 085B          ; ORG 085BH;GETL3
10 085B          ;
11 085B CDF302   GETL3: CALL    .MANG      ; CR
12 085E 0628     LD      B,40        ; 1LINE
13 0860 30C9     JR      NC,GETLA
14 0862 25       DEC     H          ; BEFORE LINE
15 0863 0650     LD      B,80        ; 2 LINE
16 0865 2E00     LD      L,0
17 0867 CDB40F   CALL    ?PNT1
18 086A D1       POP     DE          ; STORE TOP ADR.
19 086B D5       PUSH    DE
20 086C 7E       LD      A,(HL)
21 086D CDCE0B   CALL    ?DACN
22 0870 12       LD      (DE),A
23 0871 23       INC     HL
24 0872 13       INC     DE
25 0873 10F7     DJNZ    GETLZ
26 0875 EB       EX      DE,HL
27 0876 360D     LD      (HL),0DH
28 0878 2B       DEC     HL
29 0879 7E       LD      A,(HL)
30 087A FE20     CP      20H        ; SPACE THEN CR
31 087C          ;
32 087C          ;
33 087C          ; CR AND NEW LINE
34 087C          ;
35 087C 28F8     JR      Z,GETLU
36 087E          ;
37 087E          ; NEW LINE RETURN
38 087E          ;
39 087E CD0E09   GETLR: CALL    ?LTNL
40 0881 D1       POP     DE
41 0882 E1       POP     HL
42 0883 C1       POP     BC
43 0884 F1       POP     AF
44 0885 C9       RET
45 0886          ;
46 0886          ;
47 0886          ;
48 0886          ; DEFS    +13
49 0893          ; ORG 0893H
50 0893          ;
51 0893          ; MESSAGE PRINT
52 0893          ;
53 0893          ; DE PRINT DATA LOW ADR.
54 0893          ; END=CR
55 0893          ;
56 0893          ; MSG: ENT
57 0893 F5       PUSH    AF
58 0894 C5       PUSH    BC
59 0895 D5       PUSH    DE
60 0896 1A       MSG1: LD      A,(DE)

```

```

01 0897 FE0D          CP      0DH          ; CR
02 0899 280C          JR      Z,MSGX2
03 089B CD3509        CALL    ?PRNT
04 089E 13            INC      DE
05 089F 18F5          JR      MSG1
06 08A1              ;
07 08A1              ;
08 08A1              ;ORG 08A1H
09 08A1              ;
10 08A1              ; ALL PRINT MESSAGE
11 08A1              ;
12 08A1              ?MSGX: ENT
13 08A1 F5            PUSH    AF
14 08A2 C5            PUSH    BC
15 08A3 D5            PUSH    DE
16 08A4 1A            LD      A,(DE)
17 08A5 FE0D          CP      0DH
18 08A7 CAE60E        JP      Z,?RSTR1
19 08AA CDB90B        CALL    ?ADCN
20 08AD CD6C09        CALL    PRNT3
21 08B0 13            INC      DE
22 08B1 18F1          JR      MSGX1
23 08B3              ;
24 08B3              ; TOP OF KEYTBLS
25 08B3              ;
26 08B3 112A0C        ?KYSM: LD      DE,KTBL5
27 08B6 1842          JR      ?KY5
28 08B8              ;
29 08B8              ; BREAK CODE IN
30 08B8              ;
31 08B8 3ECB          #BRK: LD      A,CBH          ; BREAK CODE
32 08BA B7            OR      A
33 08BB 1819          JR      ?KY1
34 08BD              ;
35 08BD              ;
36 08BD              ;ORG 08BDH
37 08BD              ;
38 08BD              ; GETKEY
39 08BD              ;
40 08BD              ; NOT ECHO BACK
41 08BD              ;
42 08BD              ; EXIT:ACC=ASCII CODE
43 08BD              ;
44 08BD              ?GET: ENT
45 08BD CDCA08        CALL    ?KEY          ; KEY IN (DISPLAY CODE)
46 08C0 D6F0          SUB     F0H          ; NOT KEYIN CODE
47 08C2 C8            RET      Z
48 08C3 C6F0          ADD     A,F0H
49 08C5 C3CE0B        JP      ?DACN        ; DIAPLAY TO ASCII CODE
50 08C8              ;
51 08C8              ;
52 08C8              ; DEFS +2
53 08CA              ;
54 08CA              ;
55 08CA              ;
56 08CA              ;
57 08CA              ;ORG 08CAH;?KEY
58 08CA              ;
59 08CA              ; 1KEY INPUT
60 08CA              ; IN      B = KEY MODE(SHIFT,CTRL,BREAK)

```

```

01 08CA              ; C = KEY DATA (COLUMN & ROW)
02 08CA              ; EXIT ACC=DISPLAY CODE
03 08CA              ; IF NO KEY ACC=F0H
04 08CA              ; IF CY=1 THEN ATTRIBUTE ON
05 08CA              ; (SMALL,HIRAKANA)
06 08CA              ;
07 08CA              ;?KEY: ENT
08 08CA C5            PUSH    BC
09 08CB D5            PUSH    DE
10 08CC E5            PUSH    HL
11 08CD CD3008        CALL    DSWEP          ; DELAY AND KEY SWEP
12 08D0 78            LD      A,B
13 08D1 07            RLCA
14 08D2 3806          JR      C,?KY2
15 08D4 3EF0          LD      A,F0H
16 08D6              ?KY1: ENT
17 08D6 E1            POP     HL
18 08D7 D1            POP     DE
19 08D8 C1            POP     BC
20 08D9 C9            RET
21 08DA              ;
22 08DA              ;?KY2: ENT
23 08DA 11EA0B        LD      DE,KTBL          ; NORMAL KEY TABLE
24 08DD 78            LD      A,B
25 08DE FE88          CP      88H          ; BREAK IN
26 08E0 28D6          JR      Z,#BRK
27 08E2 2600          LD      H,0          ; HL=ROW & COLUMN
28 08E4 69            LD      L,C
29 08E5 CB6F          BIT     5,A          ; CTRL CHECK
30 08E7 200E          JR      NZ,?KY5-3
31 08E9 3A7011        LD      A,(KANAF)      ; 0=NR.,1=GRAPH
32 08EC 0F            RRCA
33 08ED DAFE08        JP      C,?KYGRP      ; GRAPH MODE
34 08F0 78            LD      A,B          ; CTRL KEY CHECK
35 08F1 17            RLA
36 08F2 17            RLA
37 08F3 38BE          JR      C,?KYSM
38 08F5 1803          JR      ?KY5
39 08F7 11AA0C        LD      DE,KTBLC      ; CONTROL KEY TABLE
40 08FA              ?KY5: ENT
41 08FA 19            ADD     HL,DE          ; TABLE
42 08FB              ?KY55: ENT
43 08FB 7E            LD      A,(HL)
44 08FC 18D8          JR      ?KY1
45 08FE              ?KYGRP: ENT
46 08FE CB70          BIT     6,B
47 0900 2807          JR      Z,?KYGRS
48 0902 11E90C        LD      DE,KTBLG
49 0905 19            ADD     HL,DE
50 0906 37            SCF
51 0907 18F2          JR      ?KY55
52 0909              ;
53 0909 116A0C        ?KYGRS: LD      DE,KTBLGS
54 090C 18EC          JR      ?KY5
55 090E              ;
56 090E              ;
57 090E              ;
58 090E              ;
59 090E              ;
60 090E              ;ORG 090EH

```

```

01 090E      ;
02 090E      ; NEWLINE
03 090E      ;
04 090E      ?LTNL: ENT
05 090E AF    XOR    A      ; ROW POINTER
06 090F 329411 LD    (DPRNT),A ; CR
07 0912 3ECD  LD    A,CDH
08 0914 1843  JR     PRNT5
09 0916      DEFS    +2
10 0918      ;ORG 0918H
11 0918      ;
12 0918      ?NL: ENT
13 0918 3A9411 LD    A,(DPRNT)
14 091B B7    OR     A
15 091C C8    RET    Z
16 091D 18EF  JR     ?LTNL
17 091F      DEFS    +1
18 0920      ;ORG 0920H
19 0920      ;
20 0920      ; PRINT SPACE
21 0920      ;
22 0920      ?PRTS: ENT
23 0920 3E20  LD    A,20H
24 0922 1811  JR     ?PRNT
25 0924      ;
26 0924      ; PRINT TAB
27 0924      ;
28 0924      ?PRTT: ENT
29 0924 CD0C00 CALL  PRNTS
30 0927 3A9411 LD    A,(DPRNT)
31 092A B7    OR     A
32 092B C8    RET    Z
33 092C D60A  SUB    +10
34 092E 38F4  JR     C,-10
35 0930 20FA  JR     NZ,-4
36 0932      DEFS    +3
37 0935      ;ORG 0935H
38 0935      ;
39 0935      ; PRINT
40 0935      ;
41 0935      ; IN ACC = PRINT DATA (ASCII)
42 0935      ;
43 0935      ?PRNT: ENT
44 0935 FE0D  CP     0DH      ; CR
45 0937 28D5  JR     Z,?LTNL
46 0939 C5    PUSH  BC
47 093A 4F    LD    C,A
48 093B 47    LD    B,A
49 093C CD4609 CALL  ?PRT
50 093F 78    LD    A,B
51 0940 C1    POP   BC
52 0941 C9    RET
53 0942      ;
54 0942      ;
55 0942      MSGOK: ENT
56 0942 4F4B21 DEFB  'OK!'
57 0945 0D    DEFB  0DH
58 0946      ;ORG 0946H
59 0946      ;
60 0946      ; PRINT ROUTINE

```

```

01 0946      ; 1 CHA.
02 0946      ; INPUT:C=ASCII DATA (?DSP+?DPCT)
03 0946      ;
04 0946      ?PRT: ENT
05 0946 79    LD    A,C
06 0947 CDB90B CALL  ?ADCN      ; ASCII TO DISPLAY
07 094A 4F    LD    C,A
08 094B FEF0  CP     F0H
09 094D C8    RET    Z      ; ZERO=ILLEGAL DATA
10 094E E6F0  AND    F0H      ; MSD CHECK
11 0950 FEC0  CP     C0H
12 0952 79    LD    A,C
13 0953 2017  JR     NZ,PRNT3
14 0955 FEC7  CP     C7H
15 0957 3013  JR     NC,PRNT3 ; CRT EDITOR
16 0959      PRNT5: ENT
17 0959 CDDC0D CALL  ?DPCT
18 095C FEC3  CP     C3H
19 095E 280F  JR     Z,PRNT4
20 0960 FEC5  CP     C5H      ; HOME
21 0962 2803  JR     Z,PRNT2
22 0964 FEC6  CP     C6H      ; CLR
23 0966 C0    RET    NZ
24 0967 AF    XOR    A
25 0968 329411 LD    (DPRNT),A
26 096B C9    RET
27 096C      PRNT3: ENT
28 096C CDB50D CALL  ?DSP
29 096F 3A9411 LD    A,(DPRNT) ; TAB POINT+1
30 0972 3C    INC    A
31 0973 FE50  CP     +80
32 0975 38F1  JR     C,PRNT2+1
33 0977 D650  SUB    +80
34 0979 18ED  JR     PRNT2+1
35 097B      ;
36 097B      ;
37 097B      ;
38 097B      ;
39 097B      ;
40 097B      ; FLASSING BYPASS 1
41 097B      ;
42 097B      ;
43 097B 3A8E11 LD    A,(FLASH)
44 097E 186F  JR     FLAS2
45 0980      ;
46 0980      ; BREAK SUBROUTINE BYPASS 1
47 0980      ;
48 0980      ; CTRL OR NOT KEY
49 0980      ;
50 0980      ?BRK2: ENT
51 0980 CB6F  BIT    5,A      ; NOT OR CTRL
52 0982 2802  JR     Z,?BRK3 ; CTRL
53 0984 B7    OR     A      ; NOTKEY A=7FH
54 0985 C9    RET
55 0986      ;
56 0986 3E20  LD    A,20H ; CTRL D5=1
57 0988 B7    OR     A      ; ZERO FLG. CLR
58 0989 37    SCF
59 098A C9    RET
60 098B      ;

```

```

01 098B MSGSV: ENT
02 098B 46494C45 DEFM 'FILENAME? '
03 098F 4E414D45
04 0993 3F20
05 0995 0D DEFB 0DH
06 0996 ;
07 0996 ; DLY 7 MSEC
08 0996 ;
09 0996 DLY12: ENT
10 0996 C5 PUSH BC
11 0997 0615 LD B,15H ; 7143.9 micro sec
12 0999 CD4A0A CALL DLY3
13 099C 10FB DJNZ -3
14 099E C1 POP BC
15 099F C9 RET
16 09A0 ;
17 09A0 ;
18 09A0 ;
19 09A0 ; LOADING MESSAGE
20 09A0 ;
21 09A0 MSG?2: ENT
22 09A0 4C4F4144 DEFM 'LOADING '
23 09A4 494E4720
24 09A8 0D DEFB 0DH
25 09A9 ;
26 09A9 ;
27 09A9 ;
28 09A9 ; DELAY FOR LONG PULSE
29 09A9 ;
30 09A9 DLY4: ENT
31 09A9 3E73 LD A,73H ; 458.81 micro sec
32 09AB 3D DEC A
33 09AC C2AB09 JP NZ,-1
34 09AF C9 RET
35 09B0 ;
36 09B0 ;
37 09B0 DEFS +3
38 09B3 ;
39 09B3 ;
40 09B3 ;ORG 09B3H;??KEY
41 09B3 ;
42 09B3 ; KEY BOAD SEARCH
43 09B3 ; & DISPLAY CODE CONV.
44 09B3 ;
45 09B3 ; EXIT A = DISPLAY CODE
46 09B3 ; CY= GRAPH MODE
47 09B3 ; WITH CURSOR DISPLAY
48 09B3 ;
49 09B3 ??KEY: ENT
50 09B3 E5 PUSH HL
51 09B4 CD920B CALL ?SAVE
52 09B7 ENT
53 09B7 CD7E05 CALL FLKEY ; KEY
54 09BA 20FB JR NZ,KSL1 ; KEY IN THEN JUMP
55 09BC KSL2: ENT
56 09BC CD7E05 CALL FLKEY
57 09BF 28FB JR Z,KSL2 ; NOT KEY IN THEN JUMP
58 09C1 67 LD H,A
59 09C2 CD9609 CALL DLY12 ; DELAY CHATTER
60 09C5 CDCA08 CALL ?KEY

```

```

01 09C8 F5 PUSH AF
02 09C9 BC CP H ; CHATER CHECK
03 09CA E1 POP HL
04 09CB 20EF JR NZ,KSL2
05 09CD E5 PUSH HL
06 09CE F1 POP AF ; IN KEY DATA
07 09CF CDF005 CALL ?LOAD ; FLISHING DATA LOAD
08 09D2 E1 POP HL
09 09D3 C9 RET
10 09D4 ;
11 09D4 ;
12 09D4 ; CLEAR 2
13 09D4 ;
14 09D4 ;
15 09D4 AF #CLR08: ENT
16 09D5 XOR A ; CY FLG.
17 09D5 010008 #CLR8: ENT
18 09D8 LD BC,0800H
19 09D8 05 CLEAR: ENT
20 09D9 57 LD DE
21 09DA LD D,A ; BC = CLR BYTE SIZE
22 09DA 72 LD (HL),D ; A = CLR DATA
23 09DB 23 INC HL
24 09DC 0B DEC BC
25 09DD 78 LD A,B
26 09DE B1 OR C
27 09DF 20F9 JR NZ,CLEAR1
28 09E1 D1 POP DE
29 09E2 C9 RET
30 09E3 ;
31 09E3 ;
32 09E3 ;
33 09E3 ;
34 09E3 ; FLASHING 2
35 09E3 ;
36 09E3 ?FLS: ENT
37 09E3 F5 PUSH AF
38 09E4 E5 PUSH HL
39 09E5 3A02E0 LD A,(KEYPC)
40 09E8 07 RLCA
41 09E9 07 RLCA
42 09EA 388F JR C,FLAS1
43 09EC 3A9211 LD A,(FLSDT)
44 09EF FLAS2: ENT
45 09EF CDB10F CALL ?PONT ; DISPLAY POSITION
46 09F2 77 LD (HL),A
47 09F3 FLAS3: ENT
48 09F3 E1 POP HL
49 09F4 F1 POP AF
50 09F5 C9 RET
51 09F6 ;
52 09F6 ;
53 09F6 ;
54 09F6 DEFS +9
55 09FF ;
56 09FF ;
57 09FF ;ORG 09FF ; ?FLAS
58 09FF ;
59 09FF ?FLAS: ENT
60 09FF 18E2 JR ?FLS

```

```

01 0A01      ;
02 0A01      ;
03 0A01      ;
04 0A01      ; SHORT AND LONG PULSE FOR 1 BIT WRITE
05 0A01      ;
06 0A01      ; SHORT: ENT
07 0A01 F5    PUSH AF      ; 12
08 0A02 3E03   LD A,03H    ; 9
09 0A04 3203E0 LD (CSTPT),A ; $E003 PC3=1:16
10 0A07 CD5907 CALL DLY1    ; 20+18*21+20
11 0A0A CD5907 CALL DLY1    ; 20+18*21+20
12 0A0D 3E02   LD A,02H    ; 9
13 0A0F 3203E0 LD (CSTPT),A ; $E003 PC3=0:16
14 0A12 CD5907 CALL DLY1    ; 20+18*21+20
15 0A15 CD5907 CALL DLY1    ; 20+18*21+20
16 0A18 F1     POP AF      ; 11
17 0A19 C9     RET         ; 11
18 0A1A      ;
19 0A1A      ;
20 0A1A      ; LONG: ENT
21 0A1A F5    PUSH AF      ; 11
22 0A1B 3E03   LD A,03H    ; 9
23 0A1D 3203E0 LD (CSTPT),A ; 16
24 0A20 CDA909 CALL DLY4    ; 20+18*89+20
25 0A23 3E02   LD A,02H    ; 9
26 0A25 3203E0 LD (CSTPT),A ; 16
27 0A28 CDA909 CALL DLY4    ; 20+18*89+20
28 0A2B F1     POP AF      ; 11
29 0A2C C9     RET         ; 11
30 0A2D      ;
31 0A2D      ;
32 0A2D      ; DEFS +5
33 0A32      ;
34 0A32      ;
35 0A32      ; ORG 0A32H
36 0A32      ;
37 0A32      ; BREAK KEY CHECK
38 0A32      ; AND SHIFT,CTRL KEY CHECK
39 0A32      ;
40 0A32      ; EXIT BREAK ON : ZERO=1
41 0A32      ; OFF: ZERO=0
42 0A32      ; NO KEY : CY =0
43 0A32      ; KEY IN : CY =1
44 0A32      ; A D6=1 : SHIFT ON
45 0A32      ; =0 : OFF
46 0A32      ; D5=1 : CTRL ON
47 0A32      ; =0 : OFF
48 0A32      ; D4=1 : SFT+CNT ON
49 0A32      ; =0 : OFF
50 0A32      ;
51 0A32      ; ?BRK: ENT
52 0A32 3EF8   LD A,F8H    ; LINE 8SWEEP
53 0A34 3200E0 LD (KEYPA),A
54 0A37 00     NOP
55 0A38 3A01E0 LD A,(KEYPB)
56 0A3B 07     OR A
57 0A3C 1F     RRA
58 0A3D 0A8009 JP C,?BRK2  ; SHIFT ?
59 0A40 17     RLA
60 0A41 17     RLA

```

```

01 0A42 3004   JR NC,?BRK1 ; BREAK ?
02 0A44 3E40   LD A,40H    ; SHIFT D6=1
03 0A46 37     SCF
04 0A47 C9     RET
05 0A48      ;
06 0A48      ;
07 0A48 AF    ?BRK1: XOR A    ; SHIFT ?
08 0A49 C9     RET
09 0A4A      ;
10 0A4A      ;
11 0A4A      ; 320 U SEC DELAY
12 0A4A      ;
13 0A4A      ; DLY3: ENT
14 0A4A 3E52   LD A,52H    ; 331.35 micro sec
15 0A4C C36207 JP 0762H    ; JP DLY2+2
16 0A4F      ;
17 0A4F      ;
18 0A4F      ; DEFS +1
19 0A50      ;
20 0A50      ;
21 0A50      ;
22 0A50      ; ORG 0A50H ; ?SWEP
23 0A50      ;
24 0A50      ;
25 0A50      ; KEY BOAD SWEEP
26 0A50      ;
27 0A50      ; EXIT B,D7=0 NO DATA
28 0A50      ; =1 DATA
29 0A50      ; D6=0 SHIFT OFF
30 0A50      ; =1 SHIFT ON
31 0A50      ; D5=0 CTRL OFF
32 0A50      ; =1 CTRL ON
33 0A50      ; D4=0 SHIFT+CTRL OFF
34 0A50      ; =1 SHIFT+CTRL ON
35 0A50      ; C = ROW & COLOUMN
36 0A50      ; 7 6 5 4 3 2 1 0
37 0A50      ; * *
38 0A50      ;
39 0A50      ; ?SWEP: ENT
40 0A50 D5     PUSH DE
41 0A51 E5     PUSH HL
42 0A52 AF    XOR A
43 0A53 06F8   LD B,F8H
44 0A55 57     LD D,A
45 0A56 CD320A CALL ?BRK
46 0A59 2004   JR NZ,SWEP6
47 0A5B 1688   LD D,88H
48 0A5D 1814   JR SWEP9
49 0A5F      ;
50 0A5F 3005   JR NC,SWEP0
51 0A61 57     LD D,A
52 0A62 1802   JR SWEP0
53 0A64 CBFA   SET 7,D
54 0A66      ;
55 0A66 05     DEC B
56 0A67 78     LD A,B
57 0A68 3200E0 LD (KEYPA),A
58 0A6B FEFF   CP EFH      ; MAP SWEEP END ?
59 0A6D 2008   CP EFH
60 0A6D 2008   JR NZ,SWEP3

```

```

01 0A6F FEF8      CP      F8H      ; BREAK KEY ROW
02 0A71 28F3      JR      Z,SWEPO
03 0A73           SWEPO: ENT
04 0A73 42        LD      B,D
05 0A74 E1        POP     HL
06 0A75 01        POP     DE
07 0A76 C9        RET
08 0A77           ;
09 0A77           SWEPO3: ENT
10 0A77 3A01E0    LD      A,(KEYPB)
11 0A7A 2F        CPL
12 0A7B B7        OR      A
13 0A7C 28E8      JR      Z,SWEPO
14 0A7E 5F        LD      E,A
15 0A7F           SWEPO2: ENT
16 0A7F 2608      LD      H,B
17 0A81 78        LD      A,B
18 0A82 E60F      AND     0FH
19 0A84 07        RLCA
20 0A85 07        RLCA
21 0A86 07        RLCA
22 0A87 4F        LD      C,A
23 0A88 7B        LD      A,E
24 0A89 25        DEC     H
25 0A8A 0F        RRCA
26 0A8B 30FC      JR      NC,-2
27 0A8D 7C        LD      A,H
28 0A8E 81        ADD     A,C
29 0A8F 4F        LD      C,A
30 0A90 18D2      JR      SWEPO1
31 0A92           ;
32 0A92           ;
33 0A92           ; ASCII TO DISPLAY CODE TABL ;
34 0A92           ;
35 0A92           ATBL:
36 0A92           ; 00 - 0F ;
37 0A92 F0        DEFB    F0H      ; @
38 0A93 F0        DEFB    F0H      ; A
39 0A94 F0        DEFB    F0H      ; B
40 0A95 F3        DEFB    F3H      ; C
41 0A96 F0        DEFB    F0H      ; D
42 0A97 F5        DEFB    F5H      ; E
43 0A98 F0        DEFB    F0H      ; F
44 0A99 F0        DEFB    F0H      ; G
45 0A9A F0        DEFB    F0H      ; H
46 0A9B F0        DEFB    F0H      ; I
47 0A9C F0        DEFB    F0H      ; J
48 0A9D F0        DEFB    F0H      ; K
49 0A9E F0        DEFB    F0H      ; L
50 0A9F F0        DEFB    F0H      ; M
51 0AA0 F0        DEFB    F0H      ; N
52 0AA1 F0        DEFB    F0H      ; O
53 0AA2           ; 10 - 1F
54 0AA2 F0        DEFB    F0H      ; P
55 0AA3 C1        DEFB    C1H      ; Q CUR. DOWN
56 0AA4 C2        DEFB    C2H      ; R CUR. UP
57 0AA5 C3        DEFB    C3H      ; S CUR. RIGHT
58 0AA6 C4        DEFB    C4H      ; T CUR. LEFT
59 0AA7 C5        DEFB    C5H      ; U HOME
60 0AA8 C6        DEFB    C6H      ; V CLEAR

```

```

01 0AA9 F0        DEFB    F0H      ; W
02 0AAA F0        DEFB    F0H      ; X
03 0AAB F0        DEFB    F0H      ; Y
04 0AAC F0        DEFB    F0H      ; Z SEP.
05 0AAD F0        DEFB    F0H      ; [
06 0AAE F0        DEFB    F0H      ; \
07 0AAF F0        DEFB    F0H      ; ]
08 0AB0 F0        DEFB    F0H      ; ^
09 0AB1 F0        DEFB    F0H      ; _
10 0AB2           ; 20 - 2F ;
11 0AB2 00        DEFB    00H      ; SPACE
12 0AB3 61        DEFB    61H      ; !
13 0AB4 62        DEFB    62H      ; "
14 0AB5 63        DEFB    63H      ; #
15 0AB6 64        DEFB    64H      ; $
16 0AB7 65        DEFB    65H      ; %
17 0AB8 66        DEFB    66H      ; &
18 0AB9 67        DEFB    67H      ; '
19 0ABA 68        DEFB    68H      ; (
20 0ABB 69        DEFB    69H      ; )
21 0ABC 6B        DEFB    6BH      ; *
22 0ABD 6A        DEFB    6AH      ; +
23 0ABE 2F        DEFB    2FH      ; ,
24 0ABF 2A        DEFB    2AH      ; -
25 0AC0 2E        DEFB    2EH      ; .
26 0AC1 2D        DEFB    2DH
27 0AC2           ; 30 - 3F ;
28 0AC2 20        DEFB    20H      ; 0
29 0AC3 21        DEFB    21H      ; 1
30 0AC4 22        DEFB    22H      ; 2
31 0AC5 23        DEFB    23H      ; 3
32 0AC6 24        DEFB    24H      ; 4
33 0AC7 25        DEFB    25H      ; 5
34 0AC8 26        DEFB    26H      ; 6
35 0AC9 27        DEFB    27H      ; 7
36 0ACA 28        DEFB    28H      ; 8
37 0ACB 29        DEFB    29H      ; 9
38 0ACC 4F        DEFB    4FH      ; :
39 0ACD 2C        DEFB    2CH      ; ;
40 0ACE 51        DEFB    51H      ; <
41 0ACF 2B        DEFB    2BH      ; =
42 0AD0 57        DEFB    57H      ; >
43 0AD1 49        DEFB    49H      ; ?
44 0AD2           ; 40 - 4F ;
45 0AD2 55        DEFB    55H      ; @
46 0AD3 01        DEFB    01H      ; A
47 0AD4 02        DEFB    02H      ; B
48 0AD5 03        DEFB    03H      ; C
49 0AD6 04        DEFB    04H      ; D
50 0AD7 05        DEFB    05H      ; E
51 0AD8 06        DEFB    06H      ; F
52 0AD9 07        DEFB    07H      ; G
53 0ADA 08        DEFB    08H      ; H
54 0ADB 09        DEFB    09H      ; I
55 0ADC 0A        DEFB    0AH      ; J
56 0ADD 0B        DEFB    0BH      ; K
57 0ADE 0C        DEFB    0CH      ; L
58 0ADF 0D        DEFB    0DH      ; M
59 0AE0 0E        DEFB    0EH      ; N
60 0AE1 0F        DEFB    0FH      ; O

```

```

01 0AE2          ; 50 - 5F ;
02 0AE2 10      DEFB 10H      ; P
03 0AE3 11      DEFB 11H      ; Q
04 0AE4 12      DEFB 12H      ; R
05 0AE5 13      DEFB 13H      ; S
06 0AE6 14      DEFB 14H      ; T
07 0AE7 15      DEFB 15H      ; U
08 0AE8 16      DEFB 16H      ; V
09 0AE9 17      DEFB 17H      ; W
10 0AEA 18      DEFB 18H      ; X
11 0AEB 19      DEFB 19H      ; Y
12 0AEC 1A      DEFB 1AH      ; Z
13 0AED 52      DEFB 52H      ; [
14 0AEE 59      DEFB 59H      ; \
15 0AEF 54      DEFB 54H      ; ]
16 0AF0 50      DEFB 50H      ; ^
17 0AF1 45      DEFB 45H      ; _
18 0AF2          ; 60 - 6F ;
19 0AF2 C7      DEFB C7H      ; UFO
20 0AF3 C8      DEFB C8H
21 0AF4 C9      DEFB C9H
22 0AF5 CA      DEFB CAH
23 0AF6 CB      DEFB CBH
24 0AF7 CC      DEFB CCH
25 0AF8 CD      DEFB CDH
26 0AF9 CE      DEFB CEH
27 0AFA CF      DEFB CFH
28 0AFB DF      DEFB DFH
29 0AFC E7      DEFB E7H
30 0AFD E8      DEFB E8H
31 0AFE E5      DEFB E5H
32 0AFF E9      DEFB E9H
33 0B00 EC      DEFB ECH
34 0B01 ED      DEFB EDH
35 0B02          ; 70 - 7F ;
36 0B02 D0      DEFB D0H
37 0B03 D1      DEFB D1H
38 0B04 D2      DEFB D2H
39 0B05 D3      DEFB D3H
40 0B06 D4      DEFB D4H
41 0B07 D5      DEFB D5H
42 0B08 D6      DEFB D6H
43 0B09 D7      DEFB D7H
44 0B0A D8      DEFB D8H
45 0B0B D9      DEFB D9H
46 0B0C DA      DEFB DAH
47 0B0D DB      DEFB DBH
48 0B0E DC      DEFB DCH
49 0B0F DD      DEFB DDH
50 0B10 DE      DEFB DEH
51 0B11 C0      DEFB C0H
52 0B12          ; 80 - 8F ;
53 0B12 80      DEFB 80H      ; }
54 0B13 8D      DEFB 8DH      ;
55 0B14 9D      DEFB 9DH
56 0B15 B1      DEFB B1H
57 0B16 B5      DEFB B5H
58 0B17 B9      DEFB B9H
59 0B18 B4      DEFB B4H
60 0B19 9E      DEFB 9EH

```

```

01 0B1A B2      DEFB B2H      ;
02 0B1B B6      DEFB B6H      ;
03 0B1C BA      DEFB BAH      ;
04 0B1D BE      DEFB BEH      ;
05 0B1E 9F      DEFB 9FH      ;
06 0B1F B3      DEFB B3H      ;
07 0B20 B7      DEFB B7H      ;
08 0B21 BB      DEFB BBH      ;
09 0B22          ; 90 - 9F ;
10 0B22 BF      DEFB BFH      ; -
11 0B23 A3      DEFB A3H      ;
12 0B24 85      DEFB 85H      ;
13 0B25 A4      DEFB A4H      ;
14 0B26 A5      DEFB A5H      ;
15 0B27 A6      DEFB A6H      ;
16 0B28 94      DEFB 94H      ;
17 0B29 87      DEFB 87H      ;
18 0B2A 88      DEFB 88H      ;
19 0B2B 9C      DEFB 9CH      ;
20 0B2C 82      DEFB 82H      ;
21 0B2D 98      DEFB 98H      ;
22 0B2E 84      DEFB 84H      ;
23 0B2F 92      DEFB 92H      ;
24 0B30 90      DEFB 90H      ;
25 0B31 83      DEFB 83H      ;
26 0B32          ; A0 - AF ;
27 0B32 91      DEFB 91H      ;
28 0B33 81      DEFB 81H      ;
29 0B34 9A      DEFB 9AH      ;
30 0B35 97      DEFB 97H      ;
31 0B36 93      DEFB 93H      ;
32 0B37 95      DEFB 95H      ;
33 0B38 89      DEFB 89H      ;
34 0B39 A1      DEFB A1H      ;
35 0B3A AF      DEFB AFH      ;
36 0B3B 8B      DEFB 8BH      ;
37 0B3C 86      DEFB 86H      ;
38 0B3D 96      DEFB 96H      ;
39 0B3E A2      DEFB A2H      ;
40 0B3F AB      DEFB ABH      ;
41 0B40 AA      DEFB AAH      ;
42 0B41 8A      DEFB 8AH      ;
43 0B42          ; B0 - BF ;
44 0B42 8E      DEFB 8EH      ;
45 0B43 B0      DEFB B0H      ;
46 0B44 AD      DEFB ADH      ;
47 0B45 8D      DEFB 8DH      ;
48 0B46 A7      DEFB A7H      ;
49 0B47 A8      DEFB A8H      ;
50 0B48 A9      DEFB A9H      ;
51 0B49 8F      DEFB 8FH      ;
52 0B4A 8C      DEFB 8CH      ;
53 0B4B AE      DEFB AEH      ;
54 0B4C AC      DEFB ACH      ;
55 0B4D 9B      DEFB 9BH      ;
56 0B4E A0      DEFB A0H      ;
57 0B4F 99      DEFB 99H      ;
58 0B50 BC      DEFB BCH      ; {
59 0B51 B8      DEFB B8H
60 0B52          ; C0 - CF ;

```

```

01 0B52 40      DEFB 40H
02 0B53 3B      DEFB 3BH
03 0B54 3A      DEFB 3AH
04 0B55 70      DEFB 70H
05 0B56 3C      DEFB 3CH
06 0B57 71      DEFB 71H
07 0B58 5A      DEFB 5AH
08 0B59 3D      DEFB 3DH
09 0B5A 43      DEFB 43H
10 0B5B 56      DEFB 56H
11 0B5C 3F      DEFB 3FH
12 0B5D 1E      DEFB 1EH
13 0B5E 4A      DEFB 4AH
14 0B5F 1C      DEFB 1CH
15 0B60 5D      DEFB 5DH
16 0B61 3E      DEFB 3EH
17 0B62      ; D0 - DF ;
18 0B62 5C      DEFB 5CH
19 0B63 1F      DEFB 1FH
20 0B64 5F      DEFB 5FH
21 0B65 5E      DEFB 5EH
22 0B66 37      DEFB 37H
23 0B67 7B      DEFB 7BH
24 0B68 7F      DEFB 7FH
25 0B69 36      DEFB 36H
26 0B6A 7A      DEFB 7AH
27 0B6B 7E      DEFB 7EH
28 0B6C 33      DEFB 33H
29 0B6D 4B      DEFB 4BH
30 0B6E 4C      DEFB 4CH
31 0B6F 1D      DEFB 1DH
32 0B70 6C      DEFB 6CH
33 0B71 5B      DEFB 5BH
34 0B72      ; E0 - EF ;
35 0B72 78      DEFB 78H
36 0B73 41      DEFB 41H
37 0B74 35      DEFB 35H
38 0B75 34      DEFB 34H
39 0B76 74      DEFB 74H
40 0B77 30      DEFB 30H
41 0B78 38      DEFB 38H
42 0B79 75      DEFB 75H
43 0B7A 39      DEFB 39H
44 0B7B 4D      DEFB 4DH
45 0B7C 6F      DEFB 6FH
46 0B7D 6E      DEFB 6EH
47 0B7E 32      DEFB 32H
48 0B7F 77      DEFB 77H
49 0B80 76      DEFB 76H
50 0B81 72      DEFB 72H
51 0B82      ; F0 - FF ;
52 0B82 73      DEFB 73H
53 0B83 47      DEFB 47H
54 0B84 7C      DEFB 7CH
55 0B85 53      DEFB 53H
56 0B86 31      DEFB 31H
57 0B87 4E      DEFB 4EH
58 0B88 6D      DEFB 6DH
59 0B89 48      DEFB 48H
60 0B8A 46      DEFB 46H

```

```

01 0B8B 7D      DEFB 7DH
02 0B8C 44      DEFB 44H
03 0B8D 1B      DEFB 1BH
04 0B8E 58      DEFB 58H
05 0B8F 79      DEFB 79H
06 0B90 42      DEFB 42H
07 0B91 60      DEFB 60H
08 0B92      ;
09 0B92      ;
10 0B92      ; FLASHING DATA SAVE
11 0B92      ;
12 0B92      ?SAVE: ENT
13 0B92 219211  LD HL,FLSDT
14 0B95 36EF    LD (HL),EFH ; NOMAL CURSOR
15 0B97 3A7011  LD A,(KANAF)
16 0B9A 0F      RRCA
17 0B9B 3803    JR C,SV0-2 ; GRAPH MODE
18 0B9D 0F      RRCA
19 0B9E 3002    JR NC,SV0 ; NORMAL MODE
20 0BA0 36FF    LD (HL),FFH ; GRAPH CURSOR
21 0BA2      SV0: ENT
22 0BA2 7E      LD A,(HL)
23 0BA3 F5      PUSH AF
24 0BA4 CDB10F  CALL ?PONT ; FLASING POSITION
25 0BA7 7E      LD A,(HL)
26 0BA8 328E11  LD (FLASH),A
27 0BAB F1      POP AF
28 0BAC 77      LD (HL),A
29 0BAD AF      XOR A
30 0BAE 2100E0  LD HL,KEYPA
31 0BB1 77      LD (HL),A
32 0BB2 2F      CPL
33 0BB3 77      LD (HL),A
34 0BB4 C9      RET
35 0BB5      SV1: ENT
36 0BB5 3643    LD (HL),43H ; KANA CURSOR
37 0BB7 18E9    JR SV0
38 0BB9      ;
39 0BB9      ; ORG 0BB9H;?ADCN
40 0BB9      ;
41 0BB9      ;
42 0BB9      ; ASCII TO DISPLAY CODE CONVERTE
43 0BB9      ;
44 0BB9      ; IN ACC:ASCII
45 0BB9      ; EXIT ACC:DISPLAY CODE
46 0BB9      ;
47 0BB9      ?ADCN: ENT
48 0BB9 C5      PUSH BC
49 0BBA E5      PUSH HL
50 0BBB 21920A  LD HL,ATBL
51 0BBE 4F      LD C,A
52 0BBF 0600    LD B,0
53 0BC1 09      ADD HL,BC
54 0BC2 7E      LD A,(HL)
55 0BC3 181B    JR DACN3
56 0BC5      ;
57 0BC5 56312E30 VRNS: DEFM "V1.0A" ; VERSION MANAGEMENT
58 0BC9 41      ;
59 0BCA 0D      DEFB 0DH
60 0BCB      DEFS +3

```

```

01 0BCE      ;
02 0BCE      ;
03 0BCE      ;ORG 0BCEH;?DACN
04 0BCE      ;
05 0BCE      ;  DISPLAY CODE TO ASCII CONV.  ;
06 0BCE      ;
07 0BCE      ;  IN  ACC = DISPLAY CODE
08 0BCE      ;  EXIT ACC = ASCII
09 0BCE      ;
10 0BCE      ?DACN: ENT
11 0BCE C5    PUSH BC
12 0BCF E5    PUSH HL
13 0BD0 D5    PUSH DE
14 0BD1 21920A LD HL,ATBL
15 0BD4 54    LD D,H
16 0BD5 50    LD E,L
17 0BD6 010001 LD BC,0100H
18 0BD9 E0B1  CPIR
19 0BD8 2806  JR Z,DACN1
20 0BD0 3EF0  LD A,F0H
21 0BDF      DACN2: ENT
22 0BDF D1    POP DE
23 0BE0      DACN3: ENT
24 0BE0 E1    POP HL
25 0BE1 C1    POP BC
26 0BE2 C9    RET
27 0BE3      ;
28 0BE3      DACN1: ENT
29 0BE3 B7    OR A
30 0BE4 28    DEC HL
31 0BE5 ED52  SBC HL,DE
32 0BE7 70    LD A,L
33 0BE8 18F5  JR DACN2
34 0BEA      ;
35 0BEA      ;
36 0BEA      ;
37 0BEA      ;  KEY MATRIX TO DISPLAY CODE TABL
38 0BEA      ;
39 0BEA      KTBL: ENT
40 0BEA      ;S0 00 - 07 ;
41 0BEA BF    DEFB BFH      ; SPARE
42 0BEB CA    DEFB CAH      ; GRAPH
43 0BEC 58    DEFB 58H      ;
44 0BED C9    DEFB C9H      ; ALPHA
45 0BEE F0    DEFB F0H      ; NO
46 0BEF 2C    DEFB 2CH      ;
47 0BF0 4F    DEFB 4FH      ;
48 0BF1 CD    DEFB CDH      ; CR
49 0BF2      ;S1 08 - 0F ;
50 0BF2 19    DEFB 19H      ; Y
51 0BF3 1A    DEFB 1AH      ; Z
52 0BF4 55    DEFB 55H      ; 0
53 0BF5 52    DEFB 52H      ; [
54 0BF6 54    DEFB 54H      ; ]
55 0BF7 F0    DEFB F0H      ; NULL
56 0BF8 F0    DEFB F0H      ; NULL
57 0BF9 F0    DEFB F0H      ; NULL
58 0BFA      ;S2 0 - 17 ;
59 0BFA 11    DEFB 11H      ; Q
60 0BFB 12    DEFB 12H      ; R

```

```

01 0BFC 13    DEFB 13H      ; S
02 0BFD 14    DEFB 14H      ; T
03 0BFE 15    DEFB 15H      ; U
04 0BFF 16    DEFB 16H      ; V
05 0C00 17    DEFB 17H      ; W
06 0C01 18    DEFB 18H      ; X
07 0C02      ;S3 18 - 1F ;
08 0C02 09    DEFB 09H      ; I
09 0C03 0A    DEFB 0AH      ; J
10 0C04 0B    DEFB 0BH      ; K
11 0C05 0C    DEFB 0CH      ; L
12 0C06 0D    DEFB 0DH      ; M
13 0C07 0E    DEFB 0EH      ; N
14 0C08 0F    DEFB 0FH      ; O
15 0C09 10    DEFB 10H      ; P
16 0C0A      ;S4 20 - 27 ;
17 0C0A 01    DEFB 01H      ; A
18 0C0B 02    DEFB 02H      ; B
19 0C0C 03    DEFB 03H      ; C
20 0C0D 04    DEFB 04H      ; D
21 0C0E 05    DEFB 05H      ; E
22 0C0F 06    DEFB 06H      ; F
23 0C10 07    DEFB 07H      ; G
24 0C11 08    DEFB 08H      ; H
25 0C12      ;S5 28 - 2F ;
26 0C12 21    DEFB 21H      ; 1
27 0C13 22    DEFB 22H      ; 2
28 0C14 23    DEFB 23H      ; 3
29 0C15 24    DEFB 24H      ; 4
30 0C16 25    DEFB 25H      ; 5
31 0C17 26    DEFB 26H      ; 6
32 0C18 27    DEFB 27H      ; 7
33 0C19 28    DEFB 28H      ; 8
34 0C1A      ;S6 30 - 37 ;
35 0C1A 59    DEFB 59H      ; #
36 0C1B 50    DEFB 50H      ;
37 0C1C 2A    DEFB 2AH      ; -
38 0C1D 00    DEFB 00H      ; SPACE
39 0C1E 20    DEFB 20H      ; 0
40 0C1F 29    DEFB 29H      ; 9
41 0C20 2F    DEFB 2FH      ; .
42 0C21 2E    DEFB 2EH      ; .
43 0C22      ;S7 38 - 3F ;
44 0C22 C8    DEFB C8H      ; INST.
45 0C23 C7    DEFB C7H      ; DEL.
46 0C24 C2    DEFB C2H      ; CURSOR UP
47 0C25 C1    DEFB C1H      ; CURSOR DOWN
48 0C26 C3    DEFB C3H      ; CURSOR RIGHT
49 0C27 C4    DEFB C4H      ; CURSOR LEFT
50 0C28 49    DEFB 49H      ; ?
51 0C29 2D    DEFB 2DH      ; /
52 0C2A      ;
53 0C2A      ;  KTBL SHIFT ON
54 0C2A      ;
55 0C2A      KTBL: ENT
56 0C2A      ;S0 00-07
57 0C2A BF    DEFB BFH      ; SPARE
58 0C2B CA    DEFB CAH      ; GRAPH
59 0C2C 1B    DEFB 1BH      ; POND
60 0C2D C9    DEFB C9H      ; ALPHA

```

```

01 0C2E F0      DEFB F0H      ; NO
02 0C2F 6A      DEFB 6AH      ; +
03 0C30 6B      DEFB 6BH      ; *
04 0C31 CD      DEFB CDH      ; CR
05 0C32          ;S1 08-0F
06 0C32 99      DEFB 99H      ; y
07 0C33 9A      DEFB 9AH      ; z
08 0C34 A4      DEFB A4H      ; \
09 0C35 BC      DEFB BCH      ; (
10 0C36 40      DEFB 40H      ; )
11 0C37 F0      DEFB F0H      ; NULL
12 0C38 F0      DEFB F0H      ; NULL
13 0C39 F0      DEFB F0H      ; NULL
14 0C3A          ;S2 10-17
15 0C3A 91      DEFB 91H      ; q
16 0C3B 92      DEFB 92H      ; r
17 0C3C 93      DEFB 93H      ; s
18 0C3D 94      DEFB 94H      ; t
19 0C3E 95      DEFB 95H      ; u
20 0C3F 96      DEFB 96H      ; v
21 0C40 97      DEFB 97H      ; w
22 0C41 98      DEFB 98H      ; x
23 0C42          ;S3 18-1F
24 0C42 89      DEFB 89H      ; i
25 0C43 8A      DEFB 8AH      ; j
26 0C44 8B      DEFB 8BH      ; k
27 0C45 8C      DEFB 8CH      ; l
28 0C46 8D      DEFB 8DH      ; m
29 0C47 8E      DEFB 8EH      ; n
30 0C48 8F      DEFB 8FH      ; o
31 0C49 90      DEFB 90H      ; p
32 0C4A          ;S4 20-27
33 0C4A 81      DEFB 81H      ; a
34 0C4B 82      DEFB 82H      ; b
35 0C4C 83      DEFB 83H      ; c
36 0C4D 84      DEFB 84H      ; d
37 0C4E 85      DEFB 85H      ; e
38 0C4F 86      DEFB 86H      ; f
39 0C50 87      DEFB 87H      ; g
40 0C51 88      DEFB 88H      ; h
41 0C52          ;S5 28-2F
42 0C52 61      DEFB 61H      ; !
43 0C53 62      DEFB 62H      ; "
44 0C54 63      DEFB 63H      ; #
45 0C55 64      DEFB 64H      ; $
46 0C56 65      DEFB 65H      ; %
47 0C57 66      DEFB 66H      ; &
48 0C58 67      DEFB 67H      ; '
49 0C59 68      DEFB 68H      ; (
50 0C5A          ;S6 30-37
51 0C5A 80      DEFB 80H      ; *
52 0C5B A5      DEFB A5H      ; POND MARK
53 0C5C 2B      DEFB 2BH      ; YEN
54 0C5D 00      DEFB 00H      ; SPACE
55 0C5E 60      DEFB 60H      ;
56 0C5F 69      DEFB 69H      ; )
57 0C60 51      DEFB 51H      ; <
58 0C61 57      DEFB 57H      ; >
59 0C62          ;S7 38-3F
60 0C62 C6      DEFB C6H      ; CLR

```

```

01 0C63 C5      DEFB C5H      ; HOME
02 0C64 C2      DEFB C2H      ; CURSOR UP
03 0C65 C1      DEFB C1H      ; CURSOR DOWN
04 0C66 C3      DEFB C3H      ; CURSOR RIGHT
05 0C67 C4      DEFB C4H      ; CURSOR LEFT
06 0C68 5A      DEFB 5AH      ;
07 0C69 45      DEFB 45H      ;
08 0C6A          ;
09 0C6A          ; GRAPHIC
10 0C6A          ;
11 0C6A          ; KTBLS: ENT
12 0C6A          ;S0 00-07
13 0C6A BF      DEFB BFH      ; SPARE
14 0C6B F0      DEFB F0H      ; GRAPH BUT NULL
15 0C6C E5      DEFB E5H      ; #
16 0C6D C9      DEFB C9H      ; ALPHA
17 0C6E F0      DEFB F0H      ; NO
18 0C6F 42      DEFB 42H      ; #;
19 0C70 B6      DEFB B6H      ; #:
20 0C71 CD      DEFB CDH      ; CR
21 0C72          ;S1 08-0F
22 0C72 75      DEFB 75H      ; #Y
23 0C73 76      DEFB 76H      ; #Z
24 0C74 B2      DEFB B2H      ; #0
25 0C75 D8      DEFB D8H      ; #[
26 0C76 4E      DEFB 4EH      ; #]
27 0C77 F0      DEFB F0H      ; #NULL
28 0C78 F0      DEFB F0H      ; #NULL
29 0C79 F0      DEFB F0H      ; #NULL
30 0C7A          ;S2 10-17
31 0C7A 3C      DEFB 3CH      ; #Q
32 0C7B 30      DEFB 30H      ; #R
33 0C7C 44      DEFB 44H      ; #S
34 0C7D 71      DEFB 71H      ; #T
35 0C7E 79      DEFB 79H      ; #U
36 0C7F DA      DEFB DAH      ; #V
37 0C80 38      DEFB 38H      ; #W
38 0C81 6D      DEFB 6DH      ; #X
39 0C82          ;S3 18-1F
40 0C82 7D      DEFB 7DH      ; #I
41 0C83 5C      DEFB 5CH      ; #J
42 0C84 5B      DEFB 5BH      ; #K
43 0C85 B4      DEFB B4H      ; #L
44 0C86 1C      DEFB 1CH      ; #M
45 0C87 32      DEFB 32H      ; #N
46 0C88 B0      DEFB B0H      ; #O
47 0C89 D6      DEFB D6H      ; #P
48 0C8A          ;S4 20-27
49 0C8A 53      DEFB 53H      ; #A
50 0C8B 6F      DEFB 6FH      ; #B
51 0C8C DE      DEFB DEH      ; #C
52 0C8D 47      DEFB 47H      ; #D
53 0C8E 34      DEFB 34H      ; #E
54 0C8F 4A      DEFB 4AH      ; #F
55 0C90 4B      DEFB 4BH      ; #G
56 0C91 72      DEFB 72H      ; #H
57 0C92          ;S5 28-2F
58 0C92 37      DEFB 37H      ; #1
59 0C93 3E      DEFB 3EH      ; #2
60 0C94 7F      DEFB 7FH      ; #3

```

```

01 0C95 7B      DEFB 7BH      ; #4
02 0C96 3A      DEFB 3AH      ; #5
03 0C97 5E      DEFB 5EH      ; #6
04 0C98 1F      DEFB 1FH      ; #7
05 0C99 BD      DEFB BDH      ; #8
06 0C9A          ;S6 30-3F      ; #YEN
07 0C9A D4      DEFB D4H      ; #+
08 0C9B 9E      DEFB 9EH      ; #-
09 0C9C D2      DEFB D2H      ; SPACE
10 0C9D 00      DEFB 00H      ; #0
11 0C9E 9C      DEFB 9CH      ; #9
12 0C9F A1      DEFB A1H      ; #,
13 0CA0 CA      DEFB CAH      ; #.
14 0CA1 B8      DEFB B8H      ; INST
15 0CA2          ;S7 38-3F      ; DEL.
16 0CA2 C8      DEFB C8H      ; CURSOR UP
17 0CA3 C7      DEFB C7H      ; CURSOR DOWN
18 0CA4 C2      DEFB C2H      ; CURSOR RIGHT
19 0CA5 C1      DEFB C1H      ; CURSOR LEFT
20 0CA6 C3      DEFB C3H      ; #?
21 0CA7 C4      DEFB C4H      ; #/
22 0CA8 BA      DEFB BAH      ;
23 0CA9 DB      DEFB DBH      ;
24 0CAA          ;
25 0CAA          ; CONTROL CODE
26 0CAA          ;
27 0CAA          ; KTBLC: ENT
28 0CAA          ;S0 00-07N      ;
29 0CAA F0      DEFB F0H      ;
30 0CAB F0      DEFB F0H      ;
31 0CAC F0      DEFB F0H      ;
32 0CAD F0      DEFB F0H      ;
33 0CAE F0      DEFB F0H      ;
34 0CAF F0      DEFB F0H      ;
35 0CB0 F0      DEFB F0H      ;
36 0CB1 F0      DEFB F0H      ;
37 0CB2          ;S1 08-0F      ;
38 0CB2 F0      DEFB F0H      ; Y E3
39 0CB3 5A      DEFB 5AH      ; Z E4 (CHECKER)
40 0CB4 F0      DEFB F0H      ; @
41 0CB5 F0      DEFB F0H      ; [ E5
42 0CB6 F0      DEFB F0H      ; J E7
43 0CB7 F0      DEFB F0H      ;
44 0CB8 F0      DEFB F0H      ;
45 0CB9 F0      DEFB F0H      ;
46 0CBA          ;S2 10-17      ;
47 0CBA C1      DEFB C1H      ; Q
48 0CBB C2      DEFB C2H      ; R
49 0CBC C3      DEFB C3H      ; S
50 0CBD C4      DEFB C4H      ; T
51 0CBE C5      DEFB C5H      ; U
52 0CBF C6      DEFB C6H      ; V
53 0CC0 F0      DEFB F0H      ; W E1
54 0CC1 F0      DEFB F0H      ; X E2
55 0CC2          ;S3 18-1F      ;
56 0CC2 F0      DEFB F0H      ; I F9
57 0CC3 F0      DEFB F0H      ; J FA
58 0CC4 F0      DEFB F0H      ; K FB
59 0CC5 F0      DEFB F0H      ; L FC
60 0CC6 F0      DEFB F0H      ; M FD

```

```

01 0CC7 F0      DEFB F0H      ; N F1
02 0CC8 F0      DEFB F0H      ; O F2
03 0CC9 F0      DEFB F0H      ; P F3
04 0CCA          ;S4 20-27      ;
05 0CCA F0      DEFB F0H      ; A F1
06 0CCB F0      DEFB F0H      ; B F2
07 0CCC F0      DEFB F0H      ; C F3
08 0CCD F0      DEFB F0H      ; D F4
09 0CCE F0      DEFB F0H      ; E F5
10 0CCF F0      DEFB F0H      ; F F6
11 0CD0 F0      DEFB F0H      ; G F7
12 0CD1 F0      DEFB F0H      ; H F8
13 0CD2          ;S5 28-2F      ;
14 0CD2 F0      DEFB F0H      ;
15 0CD3 F0      DEFB F0H      ;
16 0CD4 F0      DEFB F0H      ;
17 0CD5 F0      DEFB F0H      ;
18 0CD6 F0      DEFB F0H      ;
19 0CD7 F0      DEFB F0H      ;
20 0CD8 F0      DEFB F0H      ;
21 0CD9 F0      DEFB F0H      ;
22 0CDA          ;S6 30-37      ;
23 0CDA F0      DEFB F0H      ; YEN E6
24 0CDB F0      DEFB F0H      ;
25 0CDC F0      DEFB F0H      ;
26 0CDD F0      DEFB F0H      ;
27 0CDE F0      DEFB F0H      ;
28 0CDF F0      DEFB F0H      ; , EF
29 0CE0 F0      DEFB F0H      ;
30 0CE1          ;S7 38-3F      ;
31 0CE1 F0      DEFB F0H      ;
32 0CE2 F0      DEFB F0H      ;
33 0CE3 F0      DEFB F0H      ;
34 0CE4 F0      DEFB F0H      ;
35 0CE5 F0      DEFB F0H      ;
36 0CE6 F0      DEFB F0H      ;
37 0CE7 F0      DEFB F0H      ;
38 0CE8 F0      DEFB F0H      ; / EE
39 0CE9          ;
40 0CE9          ; KANA
41 0CE9          ;
42 0CE9          ; KTBLC: ENT
43 0CE9          ;S0 00-07      ;
44 0CE9 BF      DEFB BFH      ; SPARE
45 0CEA F0      DEFB F0H      ; GRAPH BUT NULL
46 0CEB CF      DEFB CFH      ; NIKO WH.
47 0CEC C9      DEFB C9H      ; ALPHA
48 0CED F0      DEFB F0H      ; NO
49 0CEE B5      DEFB B5H      ; MO
50 0CEF 4D      DEFB 4DH      ; DAKU TEN
51 0CF0 CD      DEFB CDH      ; CR
52 0CF1          ;S1 08-0F      ;
53 0CF1 35      DEFB 35H      ; HA
54 0CF2 77      DEFB 77H      ; TA
55 0CF3 D7      DEFB D7H      ; WA
56 0CF4 B3      DEFB B3H      ; YO
57 0CF5 B7      DEFB B7H      ; HANDAKU
58 0CF6 F0      DEFB F0H      ;
59 0CF7 F0      DEFB F0H      ;
60 0CF8 F0      DEFB F0H      ;

```

```

01 0CF9          ;S2      10-17
02 0CF9 7C      DEFB 7CH      ; KA
03 0CFA 70      DEFB 70H      ; KE
04 0CFB 41      DEFB 41H      ; SHI
05 0CFC 31      DEFB 31H      ; KO
06 0CFD 39      DEFB 39H      ; HI
07 0CFE A6      DEFB A6H      ; TE
08 0CFF 78      DEFB 78H      ; KI
09 0D00 0D      DEFB 0DH      ; CHI
10 0D01          ;S3      18-1F
11 0D01 3D      DEFB 3DH      ; FU
12 0D02 5D      DEFB 5DH      ; MI
13 0D03 6C      DEFB 6CH      ; MU
14 0D04 56      DEFB 56H      ; ME
15 0D05 1D      DEFB 1DH      ; RHI
16 0D06 33      DEFB 33H      ; RA
17 0D07 05      DEFB 05H      ; HE
18 0D08 B1      DEFB B1H      ; HO
19 0D09          ;S4      20-27
20 0D09 46      DEFB 46H      ; SA
21 0D0A 6E      DEFB 6EH      ; TO
22 0D0B 09      DEFB 09H      ; THU
23 0D0C 48      DEFB 48H      ; SU
24 0D0D 74      DEFB 74H      ; KU
25 0D0E 43      DEFB 43H      ; SE
26 0D0F 4C      DEFB 4CH      ; SO
27 0D10 73      DEFB 73H      ; MA
28 0D11          ;S5      28-2F
29 0D11 3F      DEFB 3FH      ; A
30 0D12 36      DEFB 36H      ; I
31 0D13 7E      DEFB 7EH      ; U
32 0D14 3B      DEFB 3BH      ; E
33 0D15 7A      DEFB 7AH      ; O
34 0D16 1E      DEFB 1EH      ; NA
35 0D17 5F      DEFB 5FH      ; NI
36 0D18 A2      DEFB A2H      ; NU
37 0D19          ;S6      30-37
38 0D19 03      DEFB 03H      ; YO
39 0D1A 9F      DEFB 9FH      ; YU
40 0D1B 01      DEFB 01H      ; YA
41 0D1C 00      DEFB 00H      ; SPACE
42 0D1D 9D      DEFB 9DH      ; NO
43 0D1E A3      DEFB A3H      ; NE
44 0D1F D0      DEFB D0H      ; RU
45 0D20 B9      DEFB B9H      ; RE
46 0D21          ;S7      38-3F
47 0D21 C6      DEFB C6H      ; ?CLR
48 0D22 C5      DEFB C5H      ; ?HOME
49 0D23 C2      DEFB C2H      ; ?CURSOR UP
50 0D24 C1      DEFB C1H      ; ?CURSOR DOWN
51 0D25 C3      DEFB C3H      ; ?CURSOR RIGHT
52 0D26 C4      DEFB C4H      ; ?CURSOR LEFT
53 0D27 BB      DEFB BBH      ; DASH
54 0D28 BE      DEFB BEH      ; RO
55 0D29          ;
56 0D29          ; MEMORY DUMP
57 0D29          ; COMMAND 'D'
58 0D29          ;
59 0D29          ; DUMP: ENT
60 0D29 CD3D01  CALL HEX1Y      ; START ADR.

```

```

01 0D2C CDA602  CALL .4DE
02 0D2F E5      PUSH HL
03 0D30 CD1004  CALL HLHEX      ; END ADR.
04 0D33 01      POP DE
05 0D34 3852    JR C,DUM1      ; DATA ER. THEN
06 0D36 E8      EX DE,HL
07 0D37 0608    DUM3: LD B,08H      ; DISP 8BYTES
08 0D39 0E17    LD C,23      ; CHA. PRINT BIAS
09 0D3B CDA05   CALL NLPHL      ; NEWLINE PRINT
10 0D3E CDB103  DUM2: CALL SPHEX      ; SP. PRT.+ACC PRT.
11 0D41 23      HL
12 0D42 F5      PUSH AF
13 0D43 3A7111  LD A,(DSPXY)      ; DISPLAY POINT
14 0D46 81      ADD A,C
15 0D47 327111  LD (DSPXY),A      ; X AXIS.=X+Creg
16 0D4A F1      POP AF
17 0D4B FE20    CP 20H
18 0D4D 3002    JR NC,+4
19 0D4F 3E2E    LD A,2EH      ; '.'
20 0D51 CDB90B  CALL ?ADCN      ; ASCII TO DISPLAY CODE
21 0D54 CD6C09  CALL PRNT3
22 0D57 3A7111  LD A,(DSPXY)
23 0D5A 0C      INC C
24 0D5B 91      SUB C      ; ASCII DSP POSITION
25 0D5C 327111  LD (DSPXY),A
26 0D5F 0D      DEC C
27 0D60 0D      DEC C
28 0D61 0D      DEC C
29 0D62 E5      PUSH HL
30 0D63 ED52    SBC HL,DE
31 0D65 E1      POP HL
32 0D66 281D    JR Z,DUM1-3
33 0D68 3EF8    LD A,F8H
34 0D6A 3200E0  LD (KEYPA),A
35 0D6D 00      NOP
36 0D6E 3A01E0  LD A,(KEYPB)
37 0D71 FEFE    CP FEH      ; SHIFT KEY ?
38 0D73 2003    JR NZ,+5
39 0D75 CDA60D  CALL ?BLNK      ; 64MSEC DELAY
40 0D78 10C4    DJNZ DUM2
41 0D7A CDCA08  CALL ?KEY      ; STOP DISP
42 0D7D B7      OR A
43 0D7E 28FA    JR Z,-4
44 0D80 CD320A  CALL ?BRK      ; SPACE KEY THEN STOP
45 0D83 20B2    JR NZ,DUM3      ; BREAK IN ?
46 0D85 C3AD00  JP ST1      ; COMMAND IN !
47 0D88 21A000  DUM1: LD HL,160      ; 20*8 BYTE
48 0D8B 19      ADD HL,DE
49 0D8C 18A8    JR DUM3-1
50 0D8E          ;
51 0D8E          ;
52 0D8E          ;
53 0D8E          ;
54 0D8E          ; DEFS +24
55 0DA6          ;
56 0DA6          ;
57 0DA6          ;
58 0DA6          ; ORG 0DA6H;?BLNK
59 0DA6          ;
60 0DA6          ;

```

```

01 0DA6          ; V-BLANK CHECK ;
02 0DA6          ;
03 0DA6          ?BLNK: ENT
04 0DA6 F5        PUSH AF
05 0DA7 3A02E0    LD A,(KEYPC) ; V-BLNK
06 0DA8 07        RLCA
07 0DA8 30FA      JR NC,-4
08 0DA9 3A02E0    LD A,(KEYPC)
09 0DB0 07        RLCA
10 0DB1 38FA      JR C,-4
11 0DB3 F1        POP AF
12 0DB4 C9        RET
13 0DB5          ;
14 0DB5          ;ORG 0DB5H;?DSP
15 0DB5          ;
16 0DB5          ;
17 0DB5          ;
18 0DB5          ; DISPLAY ON POINTER ;
19 0DB5          ;
20 0DB5          ; ACC = DISPLAY CODE
21 0DB5          ; EXCEPT F0H
22 0DB5          ;
23 0DB5          ?DSP: ENT
24 0DB5 F5        PUSH AF
25 0DB6 C5        PUSH BC
26 0DB7 D5        PUSH DE
27 0DB8 E5        PUSH HL
28 0DB9          DSP01: ENT
29 0DB9 CDB10F    CALL ?PONT ; DISPLAY POSITION
30 0DBC 77        LD (HL),A
31 0DBD 2A7111    LD HL,(DSPXY)
32 0DC0 7D        LD A,L
33 0DC1 FE27      CP +39
34 0DC3 200B      JR NZ,DSP04
35 0DC5 CDF302    CALL .MANG
36 0DC8 3806      JR C,DSP04
37 0DCA EB        EX DE,HL
38 0DCB 3601      LD (HL),+1 ; LOGICAL 1ST COLUMN
39 0DCC 23        INC HL
40 0DCE 3600      LD (HL),0 ; LOGICAL 2ND COLUMN
41 0DD0          DSP04: ENT
42 0DD0 3EC3      LD A,C3H ; CURSL
43 0DD2 180C      JR ?DPCT+4
44 0DD4          ;
45 0DD4          ;
46 0DD4          ;
47 0DD4          ;
48 0DD4          ; GRAPHIC STATUS CHECK
49 0DD4          ;
50 0DD4 3A7011    GRSTAS: LD A,(KANAF)
51 0DD7 FE01      CP 01H
52 0DD9 3ECA      LD A,CAH
53 0DDB C9        RET
54 0DDC          ;
55 0DDC          ;
56 0DDC          ;
57 0DDC          ;
58 0DDC          ;
59 0DDC          ;
60 0DDC          ;ORG 0DDCH;?DPCT

```

```

01 0DDC          ;
02 0DDC          ;
03 0DDC          ; DISPLAY CONTROL ;
04 0DDC          ;
05 0DDC          ; ACC = CONTROL CODE
06 0DDC          ;
07 0DDC          ?DPCT: ENT
08 0DDC F5        PUSH AF
09 0DDD C5        PUSH BC
10 0DDE D5        PUSH DE
11 0DDF E5        PUSH HL
12 0DE0 47        LD B,A
13 0DE1 E6F0      AND F0H
14 0DE3 FEC0      CP C0H
15 0DE5 201B      JR NZ,CURS5
16 0DE7 A8        XOR B
17 0DE8 07        RLCA
18 0DE9 4F        LD C,A
19 0DEA 0600      LD B,+0
20 0DEC 21AA0E    LD HL,CTBL ; PAGE MODE1
21 0DEF 09        ADD HL,BC
22 0DF0 5E        LD E,(HL)
23 0DF1 23        INC HL
24 0DF2 56        LD D,(HL)
25 0DF3 2A7111    LD HL,(DSPXY)
26 0DF6 EB        EX DE,HL
27 0DF7 E9        JP (HL)
28 0DF8          ;
29 0DF8          ;
30 0DF8          ;
31 0DF8          CURSD: ENT
32 0DF8 EB        EX DE,HL ; LD HL,(DSPXY)
33 0DF9 7C        LD A,H
34 0DFA FE18      CP +24
35 0DFC 2825      JR Z,CURS4
36 0DFE 24        INC H
37 0DFF          CURS1: ENT
38 0DFF          ;
39 0DFF          ;
40 0DFF          ;
41 0DFF          CURS3: ENT
42 0DFF 227111    LD (DSPXY),HL
43 0E02 C3E50E    JP ?RSTR
44 0E05          ;
45 0E05          CURSU: ENT
46 0E05 EB        EX DE,HL ; LD HL,(DSPXY)
47 0E06 7C        LD A,H
48 0E07 B7        OR A
49 0E08 28F8      JR Z,CURS5
50 0E0A 25        DEC H
51 0E0B          CURSU1: ENT
52 0E0B 18F2      JR CURS3
53 0E0D          CURSR: ENT
54 0E0D EB        EX DE,HL ; LD HL,(DSPXY)
55 0E0E 7D        LD A,L
56 0E0F FE27      CP +39
57 0E11 3003      JR NC,CURS2
58 0E13 2C        INC L
59 0E14 18E9      JR CURS3
60 0E16          CURS2: ENT

```

```

01 0E16 2E00      LD      L,+0
02 0E18 24        INC     H
03 0E19 7C        LD      A,H
04 0E1A FE19      CP      +25
05 0E1C 38E1      JR      C,CURS1
06 0E1E 2618      LD      H,+24
07 0E20 227111    LD      (DSPXY),HL
08 0E23           CURS4: ENT
09 0E23 1848      JR      SCROL
10 0E25           ;
11 0E25           CURSL: ENT
12 0E25 EB        EX      DE,HL      ; LD HL,(DSPXY)
13 0E26 7D        LD      A,L
14 0E27 B7        OR      A
15 0E28 2803      JR      Z,+5
16 0E2A 2D        DEC     L
17 0E2B 1802      JR      CURS3
18 0E2D 2E27      LD      L,+39
19 0E2F 25        DEC     H
20 0E30 F20B0E    JP      P,CURSUI
21 0E33 2600      LD      H,0
22 0E35 227111    LD      (DSPXY),HL
23 0E38 18C8      JR      CURS5
24 0E3A           ;
25 0E3A           CLRS: ENT
26 0E3A          LD      HL,MANG
27 0E3D 0618      LD      B,27
28 0E3F CDD80F    CALL    ?CLER
29 0E42 2100D0    LD      HL,D000H      ; SCRN TOP
30 0E45 CDD409    CALL    #CLR08
31 0E48 3E71      LD      A,71H      ; COLOR DATA
32 0E4A CDD509    CALL    #CLR8      ; D800H-DFFFH CLR.
33 0E4D           HOME: ENT
34 0E4D          LD      HL,0      ; DSPXY:0 X=0,Y=0
35 0E50 18AD      JR      CURS3
36 0E52           ;
37 0E52           DEFS    +8
38 0E5A           ;
39 0E5A           CR
40 0E5A           ;
41 0E5A           CR: ENT
42 0E5A          CALL    .MANG
43 0E5D 0F        RRCA
44 0E5E 30B6      JR      NC,CURS2
45 0E60 2E00      LD      L,0
46 0E62 24        INC     H
47 0E63 FE18      CP      +24
48 0E65 2803      JR      Z,CR1
49 0E67 24        INC     H
50 0E68 1895      JR      CURS1
51 0E6A           CR1: ENT
52 0E6A          LD      (DSPXY),HL
53 0E6D           ;
54 0E6D           SCROL
55 0E6D           ;
56 0E6D           SCROL: ENT
57 0E6D          LD      BC,03C0H      ; TOP OF %CRT ADR.
58 0E70 1100D0    LD      DE,SCRN      ; 1 COLUMN
59 0E73 2128D0    LD      HL,SCRN+40    ; 1000 STORE
60 0E76 C5        PUSH    BC

```

```

01 0E77 EDB0      LDIR
02 0E79 C1        POP     BC
03 0E7A 05        PUSH    DE
04 0E7B 1100D8    LD      DE,SCRN+800H ; COLOR RAM SCROLL
05 0E7E 2128D8    LD      HL,SCRN+828H ; SCROLL TOP + 40
06 0E81 E0B0      LDIR
07 0E83 0628      LD      B,40      ; ONE LINE
08 0E85 EB        EX      DE,HL
09 0E86 3E71      LD      A,71H      ; COLOR RAM INITIAL DATA
10 0E88 CDD00F    CALL    ?DINT
11 0E8B E1        POP     HL
12 0E8C 0628      LD      B,40
13 0E8E CDD80F    CALL    ?CLER      ; LAST LINE CLEAR
14 0E91 011A00    LD      BC,26      ; ROW NUMBER +1
15 0E94 117311    LD      DE,MANG      ; LOGICAL MANAGEMENT
16 0E97 217411    LD      HL,MANG+1
17 0E9A EDB0      LDIR
18 0E9C 3600      LD      (HL),0
19 0E9E 3A7311    LD      A,(MANG)
20 0EA1 B7        OR      A
21 0EA2 2841      JR      Z,?RSTR
22 0EA4 217211    LD      HL,DSPXY+1
23 0EA7 35        DEC     (HL)
24 0EA8 18C3      JR      SCROL
25 0EAA           ;
26 0EAA           ; CONTROL CODE TABLE
27 0EAA           ;
28 0EAA           CTBL: ENT
29 0EAA          DEFW    SCROL      ; SCROLLING
30 0EAC F80D      DEFW    CURSD      ; CURSOR
31 0EAE 050E      DEFW    CURSU
32 0EB0 0D0E      DEFW    CURSR
33 0EB2 250E      DEFW    CURSL
34 0EB4 400E      DEFW    HOME
35 0EB6 3A0E      DEFW    CLRS
36 0EB8 F80E      DEFW    DEL
37 0EBA 380F      DEFW    INST
38 0EBC E10E      DEFW    ALPHA
39 0EBE EE0E      DEFW    KANA
40 0EC0 E50E      DEFW    ?RSTR
41 0EC2 E50E      DEFW    ?RSTR
42 0EC4 5A0E      DEFW    CR
43 0EC6 E50E      DEFW    ?RSTR
44 0EC8 E50E      DEFW    ?RSTR
45 0ECA           ;
46 0ECA           ;
47 0ECA           ; INST BYPASS
48 0ECA           ;
49 0ECA           ;
50 0ECA          INST2: SET    3,H      ; COLOR RAM
51 0ECC 7E        LD      A,(HL)      ; FROM
52 0ECD 23        INC     HL
53 0ECE 77        LD      (HL),A      ; TO
54 0ECF 2B        DEC     HL      ; ADR ADJ.
55 0ED0 CB9C      RES     3,H
56 0ED2 EDA8      LDD     A,C      ; CHA. TRNS.
57 0ED4 79        LD      B,A
58 0ED5 B0        OR      B      ; BC=0 ?
59 0ED6 20F2      JR      NZ,INST2
60 0ED8 EB        EX      DE,HL

```

```

01 0ED9 3600      LD      (HL),0
02 0EDB CBDC      SET     3,H      ; COLOR RAM
03 0EDD 3671      LD      (HL),71H
04 0EDF 1804      JR      ?RSTR
05 0EE1           ;
06 0EE1           ;
07 0EE1           ;
08 0EE1           ;
09 0EE1           ; ORG 0EE1H;ALPHA
10 0EE1           ;
11 0EE1           ALPHA: ENT
12 0EE1 AF        XOR      A
13 0EE2           ALPH1: ENT
14 0EE2 327011    LD      (KANAF),A
15 0EE5           ;
16 0EE5           ;
17 0EE5           ; RESTORE ;
18 0EE5           ;
19 0EE5           ?RSTR: ENT
20 0EE5 E1        POP      HL
21 0EE6           ?RSTR1: ENT
22 0EE6 D1        POP      DE
23 0EE7 C1        POP      BC
24 0EE8 F1        POP      AF
25 0EE9 C9        RET
26 0EEA           ;
27 0EEA           ; MONITOR WORK AREA ;
28 0EEA           ;
29 0000 P        SCRN: EQU   D000H
30 0003 P        KANST: EQU  E003H      ; KANA STATUS PORT
31 0EEA           ;
32 0EEA           ;
33 0EEA           ;
34 0EEA           DEFS     +4
35 0EEE           ; ORG 0EEEH;KANA
36 0EEE           ;
37 0EEE           KANA: ENT
38 0EEE CDD40D    CALL     GRSTAS
39 0EF1 CAB90D    JP      Z,DSP01      ; NOT GRAPH KEY THEN JUM
40 0EF4 3E01      LD      A,+1
41 0EF6 18EA      JR      ALPH1
42 0EF8           ;
43 0EF8           ;
44 0EF8           DEL: ENT
45 0EF8 EB        EX      DE,HL      ; LD HL,(DSPXY)
46 0EF9 7C        LD      A,H      ; HOME ?
47 0EFA B5        OR      L
48 0EFB 28E8      JR      Z,?RSTR
49 0EFD 7D        LD      A,L
50 0EFE B7        OR      A
51 0EFF 200D      JR      NZ,DEL1      ; LEFT SIDE ?
52 0F01 CDF302    CALL     .MANG
53 0F04 3808      JR      C,DEL1
54 0F06 CDB10F    CALL     ?PONT
55 0F09 2B        DEC     HL
56 0F0A 3600      LD      (HL),+0
57 0F0C 1825      JR      INST-5      ; JP CURSL
58 0F0E           DEL1: ENT
59 0F0E CDF302    CALL     .MANG
60 0F11 0F        RRCA

```

```

01 0F12 3E28      LD      A,40
02 0F14 3001      JR      NC,+3
03 0F16 07        RLCA
04 0F17 95        SUB     L      ; ACC=80
05 0F18 47        LD      B,A      ; TRNS. BYTE
06 0F19 CDB10F    CALL     ?PONT
07 0F1C 7E        LD      A,(HL)   ; CHA. FROM ADR
08 0F1D 2B        DEC     HL
09 0F1E 77        LD      (HL),A   ; TO
10 0F1F 23        INC     HL
11 0F20 CBDC      SET     3,H      ; COLOR RAM
12 0F22 7E        LD      A,(HL)
13 0F23 2B        DEC     HL
14 0F24 77        LD      (HL),A
15 0F25 CB9C      RES     3,H      ; CHA.
16 0F27 23        INC     HL
17 0F28 23        INC     HL      ; NEXT
18 0F29 10F1      DJNZ    DEL2
19 0F2B 2B        DEC     HL      ; ADR.ADJUST
20 0F2C 3600      LD      (HL),0
21 0F2E CBDC      SET     3,H
22 0F30 217100    LD      HL,71H   ; BLUE + WHITE
23 0F33 3EC4      LD      A,C4H    ; JP CURSL
24 0F35 C3E00D    JP      ?DPCT+4
25 0F38           ;
26 0F38           INST: ENT
27 0F38 CDF302    CALL     .MANG
28 0F3B 0F        RRCA
29 0F3C 2E27      LD      L,+39
30 0F3E 7D        LD      A,L
31 0F3F 3001      JR      NC,+3
32 0F41 24        INC     H
33 0F42 CDB40F    CALL     ?PNT1
34 0F45 E5        PUSH    HL
35 0F46 2A7111    LD      HL,(DSPXY)
36 0F49 3002      JR      NC,+4
37 0F4B 3E4F      LD      A,+79
38 0F4D 95        SUB     L
39 0F4E 0600      LD      B,0
40 0F50 4F        LD      C,A
41 0F51 D1        POP      DE
42 0F52 2891      JR      Z,?RSTR
43 0F54 1A        LD      A,(DE)
44 0F55 B7        OR      A
45 0F56 208D      JR      NZ,?RSTR
46 0F58 62        LD      H,D      ; HLOE
47 0F59 6B        LD      L,E
48 0F5A 2B        DEC     HL
49 0F5B C3CA0E    JP      INST2    ; JUMP NEXT (BYPASS)
50 0F5E           ;
51 0F5E           ;
52 0F5E           ; PROGRAM SAVE
53 0F5E           ;
54 0F5E           ; CMD. 'S'
55 0F5E           ;
56 0F5E           ;
57 0F5E C03D01    SAVE: ENT
58 0F61 220411    CALL     HEXIY
59 0F64 44        LD      (DTADR),HL ; START ADR.
60 0F65 4D        LD      B,H      ; DATA ADR. BUFFER

```

```

01 0F66 CDA602      CALL .4DE
02 0F69 CD3D01      CALL HEXIY      ; END ADR.
03 0F6C ED42        SBC HL,BC      ; BYTE SIZE
04 0F6E 23          INC HL
05 0F6F 220211      LD (SIZE),HL  ; BYTE SIZE BUFFER
06 0F72 CDA602      CALL .4DE
07 0F75 CD3D01      CALL HEXIY      ; EXECUTE ADR.
08 0F78 220611      LD (EXADR),HL  ; BUFFER
09 0F7B CD0900      CALL NL
10 0F7E 118B09      LD DE,MSGSV    ; SAVED FILENAME
11 0F81 DF          RST 3          ; CALL MSGX
12 0F82 CD2F01      CALL BGETL      ; FILENAME INPUT
13 0F85 CDA602      CALL .4DE
14 0F88 CDA602      CALL .4DE
15 0F8B 21F110      LD HL,NAME     ; NAME BUFFER
16 0F8E             SAV1: ENT DE
17 0F8E 13          INC DE
18 0F8F 1A          LD A,(DE)
19 0F90 77          LD (HL),A      ; FILENAME TRANS.
20 0F91 23          INC HL
21 0F92 FE0D        CP 0DH        ; END CODE
22 0F94 20F8        JR NZ,SAV1
23 0F96 3E01        LD A,01H      ; ATTRIBUTE:OBJ.
24 0F98 32F010      LD (ATRB),A
25 0F9B CD3604      CALL ?WRI
26 0F9E DA0701      JP C,?ER      ; WRITE ERROR
27 0FA1 CD7504      CALL ?WRD      ; DATA
28 0FA4 DA0701      JP C,?ER
29 0FA7 CD0900      CALL NL
30 0FAA 114209      LD DE,MSGOK    ; OK MESSAGE
31 0FAD DF          RST 3          ; CALL MSGX
32 0FAE C3AD00      JP ST1
33 0FB1             ;
34 0FB1             ;
35 0FB1             ; ORG 0FB1H;?PONT
36 0FB1             ;
37 0FB1             ;
38 0FB1             ; COMPUTE POINT ADR . ;
39 0FB1             ;
40 0FB1             ; HL = SCREEN COORDINATE
41 0FB1             ; EXIT
42 0FB1             ; HL = POINT ADR. ON SCREEN
43 0FB1             ;
44 0FB1             ?PONT: ENT
45 0FB1 2A7111      LD HL,(DSPXY)
46 0FB4             ;
47 0FB4             ; ORG 0FB4H;?PNT1
48 0FB4             ;
49 0FB4             ?PNT1: ENT
50 0FB4 F5          PUSH AF
51 0FB5 C5          PUSH BC
52 0FB6 D5          PUSH DE
53 0FB7 E5          PUSH HL
54 0FB8 C1          POP BC
55 0FB9 112800      LD DE,0028H    ; 40
56 0FBC 21D8CF      LD HL,SCRN-40
57 0FBF             ?PNT2: ENT
58 0FBF 19          ADD HL,DE
59 0FC0 05          DEC B
60 0FC1 F2BF0F      JP P,-2

```

```

01 0FC4 0600      LD B,+0
02 0FC6 09        ADD HL,BC
03 0FC7 D1        POP DE
04 0FC8 C1        POP BC
05 0FC9 F1        POP AF
06 0FCA C9        RET
07 0FCB           ;
08 0FCB           ; VERIFYING
09 0FCB           ;
10 0FCB           ; COMMAND 'V'
11 0FCB           ;
12 0FCB           VRFY: ENT
13 0FCB CD8805      CALL ?VRFY
14 0FCE DA0701      JP C,?ER
15 0FD1 114209      LD DE,MSGOK
16 0FD4 DF          RST 3
17 0FD5 C3AD00      JP ST1
18 0FD8           ;
19 0FD8           ;
20 0FD8           ;
21 0FD8           ; ORG 0FD8H;?CLER
22 0FD8           ;
23 0FD8           ;
24 0FD8           ; CLER ;
25 0FD8           ; B=SIZE
26 0FD8           ; HL=LOW ADR.
27 0FD8           ;
28 0FD8           ?CLER: ENT
29 0FD8 AF          XOR A
30 0FD9 1802        JR +4
31 0FDB           ?CLRFF: ENT
32 0FDB 3EFF        LD A,FFH
33 0FDD           ?DINT: ENT
34 0FDD 77          LD (HL),A
35 0FDE 23          INC HL
36 0FDF 10FC        DJNZ -2
37 0FE1 C9        RET
38 0FE2           ;
39 0FE2           ;
40 0FE2           ; GAP CHECK
41 0FE2           ;
42 0FE2           GAPCK: ENT
43 0FE2 C5          PUSH BC
44 0FE3 D5          PUSH DE
45 0FE4 E5          PUSH HL
46 0FE5 0101E0      LD BC,KEYPB
47 0FE8 1102E0      LD DE,CSTR
48 0FEB           GAPCK1: ENT
49 0FEB 2664        LD H,100
50 0FED           GAPCK2: ENT
51 0FED CD0106      CALL EDGE
52 0FF0 380B        JR C,GAPCK3
53 0FF2 CD4A0A      CALL DLY3
54 0FF5 1A          LD A,(DE)
55 0FF6 E620        AND 20H
56 0FF8 20F1        JR NZ,GAPCK1
57 0FFA 25          DEC H
58 0FFB 20F0        JR NZ,GAPCK2
59 0FFD           GAPCK3: ENT
60 0FFD C39B06      JP RET3

```

; CALL DLY2*3

```

01 0F66 CDA602      CALL .4DE
02 0F69 CD3D01      CALL HEXIY      ; END ADR.
03 0F6C ED42        SBC HL,BC      ; BYTE SIZE
04 0F6E 23          INC HL
05 0F6F 220211      LD (SIZE),HL  ; BYTE SIZE BUFFER
06 0F72 CDA602      CALL .4DE
07 0F75 CD3D01      CALL HEXIY      ; EXECUTE ADR.
08 0F78 220611      LD (EXADR),HL  ; BUFFER
09 0F7B CD0900      CALL NL
10 0F7E 118B09      LD DE,MSGSV    ; SAVED FILENAME
11 0F81 DF          RST 3          ; CALL MSGX
12 0F82 CD2F01      CALL BGETL      ; FILENAME INPUT
13 0F85 CDA602      CALL .4DE
14 0F88 CDA602      CALL .4DE
15 0F8B 21F110      LD HL,NAME     ; NAME BUFFER
16 0F8E             SAV1: ENT DE
17 0F8E 13          INC DE
18 0F8F 1A          LD A,(DE)
19 0F90 77          LD (HL),A      ; FILENAME TRANS.
20 0F91 23          INC HL
21 0F92 FE0D        CP 0DH        ; END CODE
22 0F94 20F8        JR NZ,SAV1
23 0F96 3E01        LD A,01H      ; ATTRIBUTE:OBJ.
24 0F98 32F010      LD (ATRB),A
25 0F9B CD3604      CALL ?WRI
26 0F9E DA0701      JP C,?ER      ; WRITE ERROR
27 0FA1 CD7504      CALL ?WRD      ; DATA
28 0FA4 DA0701      JP C,?ER
29 0FA7 CD0900      CALL NL
30 0FAA 114209      LD DE,MSGOK    ; OK MESSAGE
31 0FAD DF          RST 3          ; CALL MSGX
32 0FAE C3AD00      JP ST1
33 0FB1             ;
34 0FB1             ;
35 0FB1             ; ORG 0FB1H;?PONT
36 0FB1             ;
37 0FB1             ;
38 0FB1             ; COMPUTE POINT ADR . ;
39 0FB1             ;
40 0FB1             ; HL = SCREEN COORDINATE
41 0FB1             ; EXIT
42 0FB1             ; HL = POINT ADR. ON SCREEN
43 0FB1             ;
44 0FB1             ?PONT: ENT
45 0FB1 2A7111      LD HL,(DSPXY)
46 0FB4             ;
47 0FB4             ; ORG 0FB4H;?PNT1
48 0FB4             ;
49 0FB4             ?PNT1: ENT
50 0FB4 F5          PUSH AF
51 0FB5 C5          PUSH BC
52 0FB6 D5          PUSH DE
53 0FB7 E5          PUSH HL
54 0FB8 C1          POP BC
55 0FB9 112800      LD DE,0028H    ; 40
56 0FBC 21D8CF      LD HL,SCRN-40
57 0FBF             ?PNT2: ENT
58 0FBF 19          ADD HL,DE
59 0FC0 05          DEC B
60 0FC1 F2BF0F      JP P,-2

```

```

01 0FC4 0600      LD B,+0
02 0FC6 09        ADD HL,BC
03 0FC7 D1        POP DE
04 0FC8 C1        POP BC
05 0FC9 F1        POP AF
06 0FCA C9        RET
07 0FCB           ;
08 0FCB           ; VERIFYING
09 0FCB           ;
10 0FCB           ; COMMAND 'V'
11 0FCB           ;
12 0FCB           ; VRFY: ENT
13 0FCB CD8805      CALL ?VRFY
14 0FCE DA0701      JP C,?ER
15 0FD1 114209      LD DE,MSGOK
16 0FD4 DF          RST 3
17 0FD5 C3AD00      JP ST1
18 0FD8           ;
19 0FD8           ;
20 0FD8           ;
21 0FD8           ; ORG 0FD8H;?CLER
22 0FD8           ;
23 0FD8           ;
24 0FD8           ; CLER ;
25 0FD8           ; B=SIZE
26 0FD8           ; HL=LOW ADR.
27 0FD8           ;
28 0FD8           ; ?CLER: ENT
29 0FD8 AF          XOR A
30 0FD9 1802        JR +4
31 0FDB           ;
32 0FDB 3EFF        ?CLRFF: ENT LD A,FFH
33 0FDD           ;
34 0FDD 77          ?DINT: ENT LD (HL),A
35 0FDE 23          LD HL
36 0FDF 10FC        INC HL
37 0FE1 C9        DJNZ -2
38 0FE2           ;
39 0FE2           ;
40 0FE2           ; GAP CHECK
41 0FE2           ;
42 0FE2           ;
43 0FE2 C5          GAPCK: ENT PUSH BC
44 0FE3 D5          PUSH DE
45 0FE4 E5          PUSH HL
46 0FE5 0101E0      LD BC,KEYPB
47 0FE8 1102E0      LD DE,CSTR
48 0FEB           ;
49 0FEB 2664        GAPCK1: ENT LD H,100
50 0FED           ;
51 0FED CD0106      GAPCK2: ENT CALL EDGE
52 0FF0 380B        JR C,GAPCK3
53 0FF2 CD4A0A      CALL DLY3
54 0FF5 1A          LD A,(DE)
55 0FF6 E620        AND 20H
56 0FF8 20F1        JR NZ,GAPCK1
57 0FFA 25          DEC H
58 0FFB 20F0        JR NZ,GAPCK2
59 0FFD           ;
60 0FFD C39B06      GAPCK3: ENT JP RET3

```

; CALL DLY2*3

```

01 11A3          BUFER: ENT          ; GET LINE BUFFER
02 11A3          DEFS    +81
03 11F4          ;
04 11F4          ;
05 11F4          ;      EQU TABLE I/O PORT
06 11F4          ;
07 11F4          ;
08 E000 P        KEYPA: EQU    E000H
09 E001 P        KEYPB: EQU    E001H
10 E002 P        KEYPC: EQU    E002H
11 E003 P        KEYPF: EQU    E003H
12 E002 P        CSTR:  EQU    E002H
13 E003 P        CSTPT: EQU    E003H
14 E004 P        CONT0: EQU    E004H
15 E005 P        CONT1: EQU    E005H
16 E006 P        CONT2: EQU    E006H
17 E007 P        CONF:  EQU    E007H
18 E008 P        SUNDG: EQU    E008H
19 E008 P        TEMP:  EQU    E008H
20 11F4          ;
21 11F4          ;      END

```

```

#BRK 08B8 #CLR08 09D4 #CLR8 09D5 $MCP 006B ..LPT 017B
.LDE 02A6 .LPT 0176 .MANG 02F3 2HE1 0434 2HEX 041F
??KEY 09B3 ?ADCN 08B9 ?BEL 0577 ?BELD 0352 ?BLNK 0DA6
?BRK 0A32 ?BRK1 0A48 ?BRK2 0980 ?BRK3 0986 ?CLER 0FD8
?CLRFF 0FDB ?DACN 0BCE ?DINT 0FDD ?DPCT 0DDC ?DSP 0DB5
?ER 0107 ?FLAS 09FF ?FLS 09E3 ?GET 08BD ?GETL 07E6
?KEY 08CA ?KY1 08D6 ?KY2 08DA ?KY5 08FA ?KY55 08FB
?KYGRP 08FE ?KYGRS 0909 ?KYSM 08B3 ?LOAD 05F0 ?LTNL 090E
?MLDY 01C7 ?MODE 073E ?MSG 0893 ?MSGX 08A1 ?NL 0918
?PNT1 0FB4 ?PNT2 0FBF ?PONT 0FB1 ?PRNT 0935 ?PRT 0946
?PRTS 0920 ?PRTT 0924 ?RDD 04F8 ?RDI 04D8 ?RSTR 0EE5
?RSTR1 0EE6 ?SAVE 0B92 ?SWEP 0A50 ?TEMP 02E5 ?TMR1 0375
?TMR2 037F ?TMRD 0358 ?TMS1 0331 ?TMS2 0344 ?TMST 0308
?VRFY 0588 ?WRD 0475 ?WRI 0436 ALPH1 0EE2 ALPHA 0EE1
AMP1 119B ASC 03DA ATBL 0A92 ATRB 10F0 AUTO3 07ED
BELL 003E BGTEL 012F BRKEY 001E BUFR 11A3 CKS1 0720
CKS2 072F CKS3 0733 CKSUM 071A CLEAR 09D8 CLEAR1 09DA
CLRS 0E3A CMY0 0058 COMNT 1108 CONT0 E004 CONT1 E005
CONT2 E006 CONF E007 CR 0E5A CSM0T 1199
CSTPT E003 CSTR E002 CTBL 0EAA CURS1 00FF CURS2 0E16
CURS3 00FF CURS4 0E23 CURS5 0E02 CURSD 00F8 CURSL 0E25
CURSR 0E0D CURSU 0E05 CURSU1 0E0B DACN1 0BE3 DACN2 0BDF
DACN3 0BE0 DEL 0EF8 DEL1 0F0E DEL2 0F1C DLY1 0759
DLY12 0996 DLY2 0760 DLY3 0A4A DLY4 09A9 DPRNT 1194
DSP01 0DB9 DSP04 0DD0 DSPXY 1171 DSWEP 0830 DTADR 1104
DUM1 0D88 DUM2 0D3E DUM3 0D37 DUMP 0D29 EDG1 0607
EDG2 0613 EDGE 0601 EXADR 1106 FD 00FF FD1 0106
FD2 0102 FLAS1 097B FLAS2 09EF FLAS3 09F3 FLASH 118E
FLKEY 057E FLPST 118F FLSDT 1192 FLSS1 1191 GAP 077A
GAP1 078E GAP2 0796 GAP3 079C GAPCK 0FE2 GAPCK1 0FEB
GAPCK2 0FED GAPCK3 0FFD GETKY 001B GETL 0003 GETL1 07EA
GETL2 0818 GETL3 085B GETL5 0810 GETL6 0865 GETLA 082B
GETLB 0863 GETLC 0822 GETLR 087E GETLU 0876 GETL2 086C
GOTO 00F3 GRSTAS 0DD4 HEX 03F9 HEXIY 013D HEXJ 03E5
HL1 041D HLHEX 0410 HOME 0E4D IBUF 10F0 INST 0F38
INST2 0ECA KANA 0EEE KANAF 1170 KANST E003 KEYPA E000
KEYPB E001 KEYPC E002 KEYPF E003 KSL1 09B7 KSL2 09BC
KTBL 0BEA KTBLC 0CAA KTBLG 0CE9 KTBLGS 0C6A KTBL5 0C2A
LETNL 0006 LLPT 0470 LOA0 0116 LOAD 0111 LONG 0A1A
LPRNT 018F M#TBL 0284 MANG 1173 MCR 07A8 MCR1 07AB
MCR2 07D4 MCR3 07D7 MELDY 0030 MLD1 01D1 MLD2 0205
MLD3 02D0 MLD4 0211 MLD5 0214 MLD51 02C4 MLDSP 02BE
MLDST 02AB MONIT 0000 MOT1 06A4 MOT2 06AB MOT4 06B9
MOT5 06D8 MOT7 06B7 MOT8 06D0 MOT9 06D7 MOTOR 069F
MSG 0015 MSG#1 03FB MSG#2 03FD MSG#3 0402 MSG#7 0467
MSG1 0896 MSG#2 09A0 MSG#3 06E7 MSGE1 0147 MSGOK 0942
MSGSV 098B MSGX 0018 MSGX1 08A4 MSGX2 08A7 MST1 0705
MST2 070C MST3 0717 MSTA 0044 MSTOP 0700 MSTP 0047
MTBL 026C NAME 10F1 NL 0009 NPLHL 05FA NOADD 03E2
OCTV 11A0 ONP1 021F ONP2 022C ONP3 0265 ONPU 021C
ONTYO 119F OPTBL 029C PEN 018B PLOT 0184 PMSG 01A5
PMSG1 01A8 PRNT 0012 PRNT2 0967 PRNT3 096C PRNT4 096F
PRNT5 0959 PRNTS 000C PRNTT 000F PRTHL 03BA PRTHX 03C3
PTEST 0155 PTRN 0180 PTST0 015A PTST1 0170 RATIO 11A1
RBY1 0630 RBY2 0649 RBY3 0654 RBYTE 0624 RD1 04E6
RDA 01B6 RDDAT 002A RDINF 0027 RET1 04D2 RET2 0554
RET3 069B RTAPE 050E RTP1 0513 RTP2 0519 RTP3 0532
RTP4 0554 RTP5 0565 RTP6 0572 RTP7 056E RTP8 0553
RTP9 0574 RYTHM 02C8 SAV1 0F8E SAVE 0F5E SCRIN 0000
SCROL 0E6D SG 00F7 SHORT 0A01 SIZE 1102 SLPT 03D5

```

SP	10F0	SPHEX	03B1	SS	00A2	ST0	0070	ST1	00AD
ST2	00BB	START	004A	STRGF	1193	SUMDT	1197	SUNDG	E008
SV0	0BA2	SV1	0BB5	SWEP0	0A66	SWEP01	0A64	SWEP2	0A7F
SWEP3	0A77	SWEP6	0A5F	SWEP9	0A73	SWRK	119D	TEMP	E008
TEMPW	119E	TIMFG	119C	TIMIN	038D	TIMRD	003B	TIMST	0033
TM1	0675	TM2	0678	TM3	0688	TM4	069B	TMARK	065B
TMCNT	1195	TVF1	05B2	TVF2	05B8	TVF3	05CC	TVERFY	05AD
VERFY	002D	VGOFF	0747	VERFY	0FCB	VRNS	0BC5	WBY1	076D
WBYTE	0767	WRDAT	0024	WRI1	0444	WRI2	045E	WRI3	0464
WRINF	0021	WTAP1	0494	WTAP2	04A5	WTAP3	04D2	WTAPE	048A
XTEMP	0041								

3-2 MZ Disk control

** Z80 ASSEMBLER 22-004C <92-504M-V1.0C> PAGE 01

03.06.85

```

01 E010          ORG     E010H
02 E010          ;
03 E010          ;=====
04 E010          ;
05 E010          ;       Equate file
06 E010          ;       for MZ-800
07 E010          ;       1984.09.01
08 E010          ;=====
09 E010          ;
10 E010          ;
11 E010          ;
12 E010          ;
13 10F0 P        SP:      EQU     10F0H
14 003E P        BELL:    EQU     003EH
15 0006 P        LETNL:   EQU     0006H
16 000C P        PRNTS:   EQU     000CH
17 0012 P        PRNT:    EQU     0012H
18 0015 P        MSG:      EQU     0015H
19 001E P        BRKEY:   EQU     001EH
20 0003 P        GETL:    EQU     0003H
21 001B P        GETKY:   EQU     001BH
22 00DC P        ?DPCT:   EQU     00DCH
23 00B9 P        ?ADCN:   EQU     00B9H
24 0009 P        NL:      EQU     0009H
25 03BA P        PRTHL:   EQU     03BAH
26 03DA P        ASC:     EQU     03DAH
27 0410 P        HLHEX:   EQU     0410H
28 041F P        2HEX:    EQU     041FH
29 04D8 P        ?RDI:    EQU     04D8H
30 10F1 P        NAME:    EQU     10F1H
31 04F8 P        ?RDD:    EQU     04F8H
32 10F0 P        ATRB:    EQU     10F0H
33 1102 P        SIZE:    EQU     1102H
34 1104 P        DTADR:   EQU     1104H
35 1106 P        EXADR:   EQU     1106H
36 0027 P        RDINF:   EQU     0027H
37 002A P        RDDAT:   EQU     002AH
38 0470 P        LLPT:    EQU     0470H
39 03D5 P        SLPT:    EQU     03D5H
40 05FA P        NLPHL:   EQU     05FAH
41 03B1 P        SPHEX:   EQU     03B1H
42 0920 P        ?PRTS:   EQU     0920H
43 012F P        BGETL:   EQU     012FH
44 02A6 P        .4DE:    EQU     02A6H
45 00F3 P        GOTO:    EQU     00F3H
46 0436 P        ?WRI:    EQU     0436H
47 0475 P        ?WRD:    EQU     0475H
48 0942 P        MSGOK:   EQU     0942H
49 0588 P        ?VRFY:   EQU     0588H
50 005B P        CMY0:    EQU     005BH
51 119D P        SWRK:    EQU     119DH
52 1171 P        DSPXY:   EQU     1171H
53 E000 P        KEYPA:   EQU     E000H
54 E001 P        KEYPB:   EQU     E001H
55 08CA P        ?KEY:    EQU     08CAH
56 0A32 P        ?BRK:    EQU     0A32H
57 0147 P        MSGE1:   EQU     0147H
58 096C P        PRNT3:   EQU     096CH
59 0DA6 P        ?BLNK:   EQU     0DA6H
60 06E7 P        MSG?3:   EQU     06E7H

```

** Z80 ASSEMBLER 22-004C <92-504M-V1.0C> PAGE 02

03.06.85

```

01 001E P        BRKCK:   EQU     001EH
02 0FD8 P        ?CLER:   EQU     0FD8H
03 038D P        TIMIN:   EQU     038DH
04 119E P        TEMPW:   EQU     119EH
05 02BE P        MLDSP:   EQU     02BEH
06 0577 P        ?BEL:    EQU     0577H
07 073E P        ?MODE:   EQU     073EH
08 09D5 P        #CLR8:   EQU     09D5H
09 0308 P        ?TMST:   EQU     0308H
10 E010          ;
11 E010          ;
12 E010          ; Quick disk equ table
13 E010          ;
14 E010          ;
15 00F4 P        SIOAD:   EQU     F4H          ; Sio A Data
16 00F5 P        SIOBD:   EQU     F5H          ; Sio B Data
17 00F6 P        SIOAC:   EQU     F6H          ; Sio A Control
18 00F7 P        SIOBC:   EQU     F7H          ; Sio B Control
19 1130 P        QDTBL:   EQU     1130H        ; Iocs parameter adrs
20 1130 P        QDPA:    EQU     QDTBL        ; Iocs command
21 1131 P        QDPB:    EQU     QDPA+1        ; Iocs parameter
22 1132 P        QDPC:    EQU     QDPB+1        ; Data head adrs(1)
23 1134 P        QDPE:    EQU     QDPC+2        ; Data byte size(1)
24 1136 P        QDPG:    EQU     QDPE+2        ; Data head adrs(2)
25 1138 P        QDPI:    EQU     QDPG+2        ; Data byte size(2)
26 113A P        QDCPA:   EQU     QDPI+2        ; QD Flag byte (A)
27 113B P        QDCPB:   EQU     QDCPA+1        ; QD Flag byte (B)
28 113C P        QDCPC:   EQU     QDCPB+1        ; QD Flag byte (C)
29 113D P        HDPT:    EQU     QDCPC+1        ; Iocs work
30 113E P        HDPT0:   EQU     HDPT+1
31 113F P        FNUPS:   EQU     HDPT0+1
32 1140 P        FNUPS1:  EQU     FNUPS+1
33 1141 P        FNUPF:   EQU     FNUPS1+1
34 1142 P        FNA:     EQU     FNUPF+1
35 1143 P        FNB:     EQU     FNA+1
36 1144 P        MTF:     EQU     FNB+1
37 1145 P        RTYF:    EQU     MTF+1
38 1146 P        BFRF:    EQU     RTYF+1
39 1147 P        SYNCF:   EQU     BFRF+1
40 1148 P        RETSP:   EQU     SYNCF+1
41 1104 P        QSIZE:   EQU     1104H        ; QD File size
42 1106 P        QDTADR:  EQU     1106H        ; QD Data adrs
43 1108 P        QEXADR:  EQU     1108H        ; QD Exec adrs
44 0011 P        NAMSIZ:  EQU     0011H        ; Name size
45 CD90 P        QDIRBF:  EQU     CD90H        ; QD dir buffer
46 0001 P        OBJCD:   EQU     01H          ; Obj code
47 0002 P        BTXCD:   EQU     02H          ; Btx code
48 0003 P        BSDCD:   EQU     03H          ; Bsd code
49 0000 P        BRKCD:   EQU     00H          ; Break code
50 0028 P        NTFECD:  EQU     40           ; Not found err code
51 0029 P        HDERCD:  EQU     41           ; Hard err code
52 002A P        ALEKCD:  EQU     42           ; Already exist code
53 002E P        WPRTCD:  EQU     46           ; Write protect err code
54 0032 P        QNTRCD:  EQU     50           ; Not ready
55 0033 P        TMFECD:  EQU     51           ; Too many files err
56 0035 P        NFSECD:  EQU     53           ; No file space err code
57 0036 P        UNFMCD:  EQU     54           ; Unformat err code
58 0039 P        BDSKCD:  EQU     57           ; Bad disk err code
59 E010          ;
60 E010          SKP      H

```

```

01 E010      ;
02 E010      ;
03 E010      ;
04 E010      ;      Quick-Disk(BISYNC)
05 E010      ;      Control Program
06 E010      ;
07 E010      ;      V1.0A 04.25.1984
08 E010      ;
09 E010      ;
10 E010      ;
11 E010      ;
12 E010      ;
13 E010      ;
14 EFFF P    FMS:   EQU   EFFFH      ;INIT CHECK SIZE (60KByt
e)
15 E010      ;
16 E010      ;
17 E010      ;
18 E010      ;
19 E010      ;
20 E010      ;      QDID
21 E010      ;
22 E010      ;
23 E010      ;
24 E010      ;
25 E010      QDIOS:  ENT
26 E010 3E05      LD   A,5      ;Retry 4
27 E012 324511    LD   (RTYF),A
28 E015      ;
29 E015 F3      RTY:   DI
30 E016 CD6AE0    CALL  QMEIN
31 E019 D0      RET    NC
32 E01A F5      PUSH  AF
33 E01B FE28      CP    40
34 E01D 2849      JR    Z,RTY4
35 E01F CDE8E2    CALL  MTOF
36 E022 3A3011    LD    A,(QDPA)
37 E025 FE04      CP    4      ;Write ?
38 E027 201C      JR    NZ,RTY3
39 E029 3A4111    LD    A,(FNUPF)
40 E02C B7      OR    A
41 E02D 2816      JR    Z,RTY3
42 E02F AF      XOR    A      ;FNUPF CLR
43 E030 324111    LD    (FNUPF),A
44 E033 3A4211    LD    A,(FNA)
45 E036 E5      PUSH  HL      ;RETSP <= SP-2
46 E037 ED734811 LD    (RETSP),SP
47 E03B E1      POP   HL
48 E03C      ;
49 E03C F3      DI
50 E03D CD68E2    CALL  QDSVFN
51 E040 3820      JR    C,RTY2
52 E042 CDE8E2    CALL  MTOF
53 E045      ;
54 E045 FE03      RTY3:  CP    3
55 E047 2004      JR    NZ,RTY5
56 E049 213D11    LD    HL,HDPT
57 E04C 35      DEC   (HL)
58 E04D F1      RTY5:  POP   AF
59 E04E F5      PUSH  AF
60 E04F FE29      CP    41

```

```

01 E051 200F      JR    NZ,RTY2
02 E053      ;
03 E053 214511    LD    HL,RTYF
04 E056 35      DEC   (HL)
05 E057 2809      JR    Z,RTY2
06 E059 F1      POP   AF
07 E05A 3A4011    LD    A,(FNUPS1)
08 E05D 323F11    LD    (FNUPS),A
09 E060 18B3      JR    RTY
10 E062      ;
11 E062 CDD8E1    RTY2:  CALL  WRCAN
12 E065 CD83E0    CALL  QDHPC
13 E068 F1      RTY4:  POP   AF
14 E069 C9      RET
15 E06A      ;
16 E06A      SKP    H

```

```

01 E06A      ;
02 E06A ED734811  GMEIN: LD      (RETSP),SP
03 E06E 3A3011    LD      A,(QDPA)
04 E071 3D        DEC      A
05 E072 2816      JR       Z,QDRC      ;Ready Check
06 E074 3D        DEC      A
07 E075 2819      JR       Z,QDFM      ;Format
08 E077 3D        DEC      A
09 E078 2860      JR       Z,QDRD      ;Read
10 E07A 3D        DEC      A
11 E07B CA4EE1    JP       Z,QDWR      ;Write
12 E07E 3D        DEC      A
13 E07F 2802      JR       Z,QDHPC      ;Head Point Clear
14 E081 1850      JR       MTOFX      ;Motor Off
15 E083      ;
16 E083      ;=====
17 E083      ;
18 E083      ;      Head Point Clear
19 E083      ;
20 E083      ;=====
21 E083      ;
22 E083      ;
23 E083 F5        QDHPC: ENT      AF
24 E084 AF        PUSH     AF
25 E085 323D11    XOR       A
26 E088 F1        LD       (HDPT),A
27 E089 C9        POP      AF
28 E08A      RET
29 E08A      ;
30 E08A      ;=====
31 E08A      ;
32 E08A      ;      Ready Check
33 E08A      ;
34 E08A      ;=====
35 E08A      ;
36 E08A 3A3111    QDRC:  ENT      A,(QDPB)
37 E08D C33CE2    LD       QREDY
38 E090      ;
39 E090      SKP      H

```

```

01 E090      ;
02 E090      ;=====
03 E090      ;
04 E090      ;      Format
05 E090      ;
06 E090      ;=====
07 E090      ;
08 E090      ;
09 E090 AF        QDFM:  ENT
10 E091 CD68E2    XOR       A
11 E094 CD65E3    CALL     QDSVFN
12 E097 01FFEF    CALL     SYNC52
13 E09A 3EAA      LD       BC,FMS
14 E09C      LD       A,AAH
15 E09C 2F        ;
16 E09D 57        QDFM1: CPL
17 E09E CDD8E3    LD       D,A
18 E0A1 0B        CALL     TRANS
19 E0A2 78        DEC      BC
20 E0A3 B1        LD       A,B
21 E0A4 2803      OR       C
22 E0A6 7A        JR       Z,QDFM2
23 E0A7 18F3      LD       A,D
24 E0A9      JR       QDFM1
25 E0A9 CDB2E3    ;
26 E0AC CDE8E2    QDFM2: CALL     EOM
27 E0AF CD9BE2    CALL     MTOF
28 E0B2 3A4311    CALL     MTON
29 E0B5 3D        LD       A,(FNB)
30 E0B6 201E      DEC      A
31 E0B8 CDFDE2    JR       NZ,FMERR
32 E0BB 01FFEF    CALL     SYNCL2
33 E0BE 1E55      LD       BC,FMS
34 E0C0 8B        LD       E,55H
35 E0C1 2013      QDFM3: CP       E
36 E0C3 0B        JR       NZ,FMERR
37 E0C4 78        DEC      BC
38 E0C5 B1        LD       A,B
39 E0C6 2808      OR       C
40 E0C8 7B        JR       Z,QDFM4
41 E0C9 2F        LD       A,E
42 E0CA 5F        CPL
43 E0CB CDF0E3    LD       E,A
44 E0CE 18F0      CALL     RDATA
45 E0D0      JR       QDFM3
46 E0D0 CDC3E3    ;
47 E0D3 C3E8E2    QDFM4: CALL     RDCRC
48 E0D6      MTOFX: JP       MTOF
49 E0D6 3E29      ;
50 E0D8 37        FMERR: LD       A,41      ;Hard Err
51 E0D9 C9        SCF
52 E0DA      RET
53 E0DA      SKP      H

```

```

01 E0DA ;
02 E0DA ;=====
03 E0DA ;
04 E0DA ;           Read
05 E0DA ;
06 E0DA ;=====
07 E0DA ;
08 E0DA QDRD: ENT
09 E0DA 3A4411 LD A,(MTF)
10 E0DD B7 OR A
11 E0DE CC9BE2 CALL Z,MTON
12 E0E1 CD14E1 CALL HPS
13 E0E4 D8 RET C
14 E0E5 CD35E4 CALL BRKC
15 E0E8 ;
16 E0E8 CDF0E3 CALL RDATA
17 E0EB 4F LD C,A
18 E0EC CDF0E3 CALL RDATA
19 E0EF 47 LD B,A
20 E0F0 B1 OR C
21 E0F1 CAE5E1 JP Z,QDWE1
22 E0F4 2A3411 LD HL,(QDPE) ;Byte size check
23 E0F7 ED42 SBC HL,BC
24 E0F9 DAE5E1 JP C,QDWE1
25 E0FC 2A3211 LD HL,(QDPC)
26 E0FF ;
27 E0FF ;Block Data Read
28 E0FF ;
29 E0FF BDR: CALL RDATA
30 E102 77 LD (HL),A
31 E103 23 INC HL
32 E104 0B DEC BC
33 E105 78 LD A,B
34 E106 B1 OR C
35 E107 20F6 JR NZ,BDR
36 E109 CDC3E3 CALL RDCRC
37 E10C 3A3111 LD A,(QDPB)
38 E10F CB47 BIT 0,A
39 E111 20C0 JR NZ,MTOFX
40 E113 C9 RET
41 E114 ;
42 E114 SKP H

```

```

01 E114 ;
02 E114 ;Head Point Search
03 E114 ;
04 E114 HPS: ENT
05 E114 214311 LD HL,FNB
06 E117 35 DEC (HL)
07 E118 2830 JR Z,HPNFE ;Not Found
08 E11A CDF0E2 CALL SYNCL2
09 E11D 4F LD C,A ;BLKFLG => C reg
10 E11E 3A3D11 LD A,(HDPT)
11 E121 213E11 LD HL,HDPT0
12 E124 BE CP (HL) ;Search ok ?
13 E125 2020 JR NZ,HPS1
14 E127 3C INC A ;HDPT count up
15 E128 323D11 LD (HDPT),A
16 E12B 77 LD (HL),A ;HDPT0 count up
17 E12C 3A3111 LD A,(QDPB)
18 E12F A9 XOR C
19 E130 1F RRA
20 E131 D0 RET NC ;=
21 E132 ;
22 E132 ;Dummy read
23 E132 ;
24 E132 CDF0E3 DMR: CALL RDATA
25 E135 4F LD C,A
26 E136 CDF0E3 CALL RDATA
27 E139 47 LD B,A
28 E13A ;
29 E13A CDF0E3 DMR1: CALL RDATA
30 E13D 0B DEC BC
31 E13E 78 LD A,B
32 E13F B1 OR C
33 E140 20F8 JR NZ,DMR1
34 E142 CDC3E3 CALL RDCRC
35 E145 18CD JR HPS ;next
36 E147 ;
37 E147 34 HPS1: INC (HL)
38 E148 18E8 JR DMR
39 E14A ;
40 E14A 3E28 HPNFE: LD A,40 ;Not Found
41 E14C 37 SCF
42 E14D C9 RET
43 E14E ;
44 E14E SKP H

```

```

01 E14E      ;
02 E14E      ;=====
03 E14E      ;
04 E14E      ;      Write
05 E14E      ;
06 E14E      ;=====
07 E14E      ;
08 E14E      QDWR: ENT
09 E14E 3A3F11 LD      A,(FNUPS)
10 E151 324011 LD      (FNUPS1),A
11 E154 3A4411 LD      A,(MTF)
12 E157 B7    OR      A
13 E158 2017  JR      NZ,QDWR1
14 E15A CD9BE2 CALL   MTON
15 E15D 3A3F11 LD      A,(FNUPS)
16 E160 214311 LD      HL,FNB
17 E163 86    ADD     A,(HL)
18 E164 324311 LD      (FNB),A
19 E167 3C    INC     A
20 E168 323D11 LD      (HDPT),A
21 E16B CD14E1 CALL   HPS
22 E16E D2E5E1 JP      NC,QDWE1      ;Hard err
23 E171 3A3111 QDWR1: LD      A,(QDPB)
24 E174 47    LD      B,A
25 E175 E601  AND     1
26 E177 200C  JR      NZ,QDWR2
27 E179 113211 LD      DE,QDPC
28 E17C 78    LD      A,B
29 E17D CB97  RES     2,A
30 E17F CDEDE1 CALL   BDW
31 E182 CD35E4 CALL   BRKC
32 E185      ;
33 E185 113611 QDWR2: LD      DE,QDPG
34 E188 3A3111 LD      A,(QDPB)
35 E18B CBC7  SET     0,A
36 E18D CDEDE1 CALL   BDW
37 E190 CDE8E2 CALL   MTOF
38 E193 CD35E4 CALL   BRKC
39 E196      ;
40 E196 3A3111 LD      A,(QDPB)
41 E199 E604  AND     4
42 E19B 284B  JR      Z,NFNUP
43 E19D 3A4211 LD      A,(FNA)
44 E1A0 213F11 LD      HL,FNUPS
45 E1A3 86    ADD     A,(HL)
46 E1A4 CD68E2 CALL   QDSVFN
47 E1A7 3E01  LD      A,1
48 E1A9 324111 LD      (FNUPF),A
49 E1AC 3A4211 QDWR4: LD      A,(FNA)
50 E1AF 214011 LD      HL,FNUPS1
51 E1B2 86    ADD     A,(HL)
52 E1B3 3C    INC     A
53 E1B4 324311 LD      (FNB),A
54 E1B7 3C    INC     A
55 E1B8 323D11 LD      (HDPT),A
56 E1BB CD14E1 CALL   HPS
57 E1BE 3025  JR      NC,QDWE1
58 E1C0 3A3111 LD      A,(QDPB)
59 E1C3 E601  AND     1
60 E1C5 2007  JR      NZ,QDWR3

```

```

01 E1C7 113211 LD      DE,QDPC
02 E1CA CD1BE2 CALL   BDV
03 E1CD D8    RET     C
04 E1CE 113611 QDWR3: LD      DE,QDPG
05 E1D1 CD1BE2 CALL   BDV
06 E1D4 D8    RET     C
07 E1D5 3A4111 LD      A,(FNUPF)
08 E1D8 B7    OR      A
09 E1D9 2807  JR      Z,QDWR5
10 E1DB AF    WRCAN: XOR     A
11 E1DC 323F11 LD      (FNUPS),A
12 E1DF 324111 LD      (FNUPF),A
13 E1E2      ;
14 E1E2 C3E8E2 QDWR5: JP      MTOF
15 E1E5      ;
16 E1E5 C3D6E0 QDWE1: JP      FMERR
17 E1E8      ;
18 E1E8 CD9BE2 NFNUP: CALL   MTON
19 E1EB 18BF  JR      QDWR4
20 E1ED      ;
21 E1ED      ;Block Data Write
22 E1ED      ;
23 E1ED F5    BDW:  PUSH   AF
24 E1EE 213F11 LD      HL,FNUPS
25 E1F1 34    INC     (HL)
26 E1F2 CD65E3 CALL   SYNCS2
27 E1F5 F1    POP     AF
28 E1F6 CDDBE3 CALL   TRANS
29 E1F9 CD11E2 CALL   RSET
30 E1FC 79    LD      A,C
31 E1FD CDDBE3 CALL   TRANS
32 E200 78    LD      A,B
33 E201 CDDBE3 CALL   TRANS
34 E204      ;
35 E204 7E    BDW1:  LD      A,(HL)
36 E205 CDDBE3 CALL   TRANS
37 E208 23    INC     HL
38 E209 0B    DEC     BC
39 E20A 78    LD      A,B
40 E20B 81    OR      C
41 E20C 20F6  JR      NZ,BDW1
42 E20E C3B2E3 JP      EOM
43 E211      ;
44 E211      ;HL,BC SET
45 E211      ;
46 E211 EB    RSET:  EX      DE,HL
47 E212 5E    LD      E,(HL)
48 E213 23    INC     HL
49 E214 56    LD      D,(HL)
50 E215 23    INC     HL
51 E216 4E    LD      C,(HL)
52 E217 23    INC     HL
53 E218 46    LD      B,(HL)
54 E219 EB    EX      DE,HL
55 E21A C9    RET
56 E21B      ;
57 E21B      ;Block Data Verify
58 E21B      ;
59 E21B CDFDE2 BDV:  CALL   SYNCL2
60 E21E CD11E2 CALL   RSET

```

```

01 E221 CDF0E3      CALL  RDATA
02 E224 B9          CP      C
03 E225 20BE        JR      NZ,QDWE1
04 E227 CDF0E3      CALL  RDATA
05 E22A B8          CP      B
06 E22B 20B8        JR      NZ,QDWE1
07 E22D             ;
08 E22D CDF0E3      BDV1:  CALL  RDATA
09 E230 BE          CP      (HL)
10 E231 20B2        JR      NZ,QDWE1
11 E233 23          INC     HL
12 E234 08          DEC     BC
13 E235 78          LD      A,B
14 E236 B1          OR      C
15 E237 20F4        JR      NZ,BDV1
16 E239 C3C3E3      JP      RDCRC
17 E23C             ;
18 E23C             SKP     H

```

```

01 E23C             ;
02 E23C             ;
03 E23C             ;
04 E23C             ;
05 E23C             ; Quick-Disk
06 E23C             ;   Physical iocs (bisync)
07 E23C             ;
08 E23C             ;   By MZ- 700 IPL-ROM
09 E23C             ;   MZ-1500 IPL-ROM
10 E23C             ;   MZ- 700 Disk-Basic
11 E23C             ;   MZ-1500 Basic
12 E23C             ;
13 E23C             ;   V2.0A 04.25.1984
14 E23C             ;
15 E23C             ;
16 E23C             ;
17 E23C             ;
18 E23C             ; i/o port adrs
19 E23C             ;
20 E23C             ;SIOAD:EQU F4H; sio A data
21 E23C             ;SIOBD:EQU F5H; sio B data
22 E23C             ;SIOAC:EQU F6H; sio A control
23 E23C             ;SIOBC:EQU F7H; sio B control
24 E23C             ;
25 E23C             ;
26 E23C             ;
27 E23C             ; Ready & Write protect
28 E23C             ;   Acc = '0' : Ready check
29 E23C             ;   Acc = '1' : & Write Protect
30 E23C             ;
31 E23C             QREDY: ENT
32 E23C 47          LD      B,A
33 E23D 3E02        LD      A,02H           ;SIO hard check
34 E23F 03F7        OUT     (SIOBC),A
35 E241 3E81        LD      A,81H
36 E243 03F7        OUT     (SIOBC),A
37 E245 3E02        LD      A,02H
38 E247 03F7        OUT     (SIOBC),A
39 E249 0BF7        IN      A,(SIOBC)
40 E24B E681        AND     81H
41 E24D FE81        CP      81H
42 E24F C206E4      JP      NZ,IOE50       ;Not ready
43 E252 3E10        LD      A,10H
44 E254 03F6        OUT     (SIOAC),A
45 E256 0BF6        IN      A,(SIOAC)
46 E258 4F          LD      C,A
47 E259 E608        AND     08H
48 E25B CA06E4      JP      Z,IOE50       ;Not ready
49 E25E 78          LD      A,B
50 E25F B7          OR      A
51 E260 C8          RET      Z           ;No err
52 E261 79          LD      A,C
53 E262 E620        AND     20H
54 E264 C0          RET      NZ         ;No err
55 E265 C303E4      JP      IOE46       ;Write protect
56 E268             ;
57 E268             SKP     H

```

```

01 E268      ;
02 E268      ;
03 E268      ; Write FN
04 E268      ;
05 E268      QDSVFN: ENT
06 E268 F5    PUSH AF
07 E269 212CE4 LD HL,SIO$D
08 E26C 0609   LD B,09H
09 E26E C008E2 CALL LSINT      ;save init
10 E271      ;
11 E271 3E10   SREDY: LD A,10H
12 E273 03F6   OUT (SIOAC),A
13 E275 DBF6   IN A,(SIOAC)
14 E277 E608   AND 8
15 E279 CA06E4 JP Z,IOE50      ;Not ready
16 E27C 3E10   LD A,10H
17 E27E 03F7   OUT (SIOBC),A
18 E280 DBF7   IN A,(SIOBC)
19 E282 E608   AND 8
20 E284 28EB   JR Z,SREDY
21 E286 01E900 LD BC,00E9H      ;Wait 160ms
22 E289 CD14E4 CALL TIMW
23 E28C CD9AE3 CALL SBRK      ;Send Break
24 E28F CD7EE3 CALL SYNCS1    ;FN Only SYNC
25 E292 F1     POP AF
26 E293 C00BE3 CALL TRANS      ;FN=A
27 E296 C0B2E3 CALL EOM      ;CRC FLAG(7EH)
28 E299 1833   JR FNEND
29 E29B      ;
30 E29B      ;
31 E29B      ;
32 E29B      ; MTON -- QD MOTOR ON
33 E29B      ; READ FILE NUMBER
34 E29B      ; READ &CHECK CRC,FLAG
35 E29B      ;
36 E29B      MTON: ENT
37 E29B 2121E4 LD HL,SIO$D
38 E29E 060B   LD B,0BH
39 E2A0 C0D8E2 CALL LSINT      ;load init
40 E2A3      ;
41 E2A3 3E10   LREDY: LD A,10H
42 E2A5 03F6   OUT (SIOAC),A
43 E2A7 DBF6   IN A,(SIOAC)
44 E2A9 E608   AND 8
45 E2AB CA06E4 JP Z,IOE50      ;Not ready
46 E2AE CD35E4 CALL BRKC
47 E2B1 3E10   LD A,10H
48 E2B3 03F7   OUT (SIOBC),A
49 E2B5 DBF7   IN A,(SIOBC)
50 E2B7 E608   AND 8
51 E2B9 28E8   JR Z,LREDY
52 E2BB 01E900 LD BC,00E9H      ;Wait 160ms
53 E2BE CD14E4 CALL TIMW
54 E2C1 CD13E3 CALL SYNCL1    ;LOAD SYNC
55 E2C4 324211 LD (FNA),A
56 E2C7 3C     INC A
57 E2C8 324311 LD (FNB),A
58 E2CB CDC3E3 CALL RDCRC
59 E2CE 214711 FNEND: LD HL,SYNCF
60 E2D1 CB0E   SET 3,(HL)

```

```

01 E2D3 AF    XOR A
02 E2D4 323E11 LD (HDPT0),A
03 E2D7 C9    RET
04 E2D8      ;
05 E2D8      SKP H

```

```

01 E2D8      ;
02 E2D8      ;   sio initial
03 E2D8      ;
04 E2D8      ;
05 E2D8 0EF6  LSINT: LD      C,F6H      ; C sioac
06 E2DA EDB3      OTIR
07 E2DC 3E05      LD      A,05H
08 E2DE 324411    LD      (MTF),A
09 E2E1 D3F7      OUT     (SIOBC),A
10 E2E3 3E80      LD      A,80H
11 E2E5 D3F7      OUT     (SIOBC),A
12 E2E7 C9        RET
13 E2E8      ;
14 E2E8      ; Motor off
15 E2E8      ;
16 E2E8      ;
17 E2E8      ;
18 E2E8 F5        QDOFF: ENT
19 E2E9 3E05      MTOF:  ENT
20 E2EB D3F6      PUSH    AF
21 E2ED 3E60      LD      A,05H
22 E2EF D3F6      LD      (SIOAC),A
23 E2F1 3E05      LD      A,60H
24 E2F3 D3F7      LD      (SIOAC),A
25 E2F5 AF        LD      A,05H
26 E2F6 324411    LD      (SIOBC),A
27 E2F9 D3F7      XOR     A
28 E2FB F1        LD      (MTF),A
29 E2FC C9        LD      (SIOBC),A
30 E2FD      POP     AF
31 E2FD      RET
32 E2FD      ;
                 ; SKP      H

```

```

01 E2FD      ;
02 E2FD      ; SYNCL1 -- LOAD F.N SYNC ONLY
03 E2FD      ;   (SEND BREAK 110ms)
04 E2FD      ; SYNCL2 -- LOAD FIRST FILE SYNC
05 E2FD      ;   (SEND BREAK 110ms)
06 E2FD      ; SYNCL3 -- LOAD FILES SYNC
07 E2FD      ;   (SEND BREAK 002ms)
08 E2FD      ;
09 E2FD      ;
10 E2FD      ; SYNCL2: ENT
11 E2FD 3E58      LD      A,58H
12 E2FF 060B      LD      B,0BH
13 E301 2121E4    LD      HL,SIO LD
14 E304 C0A3E3    CALL    SYNCA
15 E307 214711    LD      HL,SYNCF
16 E30A CB5E      BIT     3,(HL)
17 E30C 010300    LD      BC,3
18 E30F 2805      JR      Z,TMLPL
19 E311 CB9E      RES     3,(HL)
20 E313 01A000    LD      BC,00A0H
21 E316          ; WAIT 2ms
22 E316          ;
23 E316 C014E4    TMLPL: CALL    TIMW
24 E319 3E05      LD      A,05H
25 E31B D3F7      OUT     (SIOBC),A
26 E31D 3E82      LD      A,82H
27 E31F D3F7      OUT     (SIOBC),A
28 E321 3E03      LD      A,03H
29 E323 D3F6      LD      (SIOAC),A
30 E325 3ED3      LD      A,D3H
31 E327 D3F6      LD      (SIOAC),A
32 E329 01C02C    LD      BC,2CC0H
33 E32C          ; loop 220ms
34 E32C          ;
35 E32C          ; SYNCL1: LD      A,10H
36 E32E D3F6      OUT     (SIOAC),A
37 E330 0BF6      LD      A,(SIOAC)
38 E332 E610      AND     10H
39 E334 2807      JR      Z,SYNCW1
40 E336 0B        DEC     BC
41 E337 78        LD      A,B
42 E338 B1        OR      C
43 E339 20F1      JR      NZ,SYNCW0
44 E33B 1817      JR      SYNW01
45 E33D          ; Un format
46 E33D          ;
47 E33D          ; SYNCL2: LD      A,03H
48 E33F D3F6      OUT     (SIOAC),A
49 E341 3EC3      LD      A,C3H
50 E343 D3F6      OUT     (SIOAC),A
51 E345 069F      LD      B,9FH
52 E347          ; loop 3ms
53 E347          ;
54 E347          ; SYNCL3: LD      A,10H
55 E349 D3F6      OUT     (SIOAC),A
56 E34B DBF6      LD      A,(SIOAC)
57 E34D E601      AND     01H
58 E34F 2006      JR      NZ,SYNCW3
59 E351 05        DEC     B
60 E352 20F3      JR      NZ,SYNCW2
61 E354 C30CE4    LD      BC,10E54
62 E357          ; Un format
63 E357          ;
64 E357          ; SYNCL1: LD      A,03H
65 E359 D3F6      OUT     (SIOAC),A
66 E35B 3EC9      LD      A,C9H

```

```

01 E35D D3F6          OUT    (SIOAC),A
02 E35F CDF0E3        CALL   RDATA
03 E362 C3F0E3        JP     RDATA
04 E365              ;
05 E365              ;
06 E365              ;
07 E365              ; SYNC1 -- SAVE F.N SYNC
08 E365              ;         (SEND BREAK 220ms)
09 E365              ; SYNC2 -- SAVE FIRST FILE SYNC
10 E365              ;         (SEND BREAK 220ms)
11 E365              ; SYNC3 -- SAVE FILES SYNC
12 E365              ;         (SEND BREAK 020ms)
13 E365              ;
14 E365              ;
15 E365 3E98          LD     A,98H
16 E367 0609          LD     B,09H
17 E369 212CE4        LD     HL,SIOSD
18 E36C CDA3E3        CALL   SYNCA
19 E36F CD9AE3        CALL   SBRK
20 E372              ;
21 E372 214711        LD     HL,SYNCF
22 E375 CB5E          BIT     3,(HL)
23 E377 011000        LD     BC,0010H      ;WAIT 20ms
24 E37A 2805          JR     Z,TMLPS
25 E37C CB9E          RES     3,(HL)
26 E37E 014001        SYNCS1: LD     BC,0140H      ;WAIT 220ms
27 E381              ;
28 E381 CD14E4        TMLPS: CALL   TIMW
29 E384 3E05          LD     A,05H
30 E386 D3F6          OUT     (SIOAC),A
31 E388 3EEF          LD     A,EFH
32 E38A D3F6          OUT     (SIOAC),A
33 E38C 010100        LD     BC,1          ;WAIT 0.7ms
34 E38F CD14E4        CALL   TIMW
35 E392 3EA5          LD     A,A5H      ; Patch (1984.10.11)
36 E394 CDDBE3        CALL   TRANS
37 E397 C380F3        JP     EOMRS      ; Eom reset & return
38 E39A              ;
39 E39A              ;
40 E39A              ;
41 E39A              ; SBRK -- SEND BREAK (00H)
42 E39A              ;
43 E39A              ;
44 E39A 3E05          SBRK:  ENT     A,05H
45 E39C D3F6          LD     OUT     (SIOAC),A
46 E39E 3EFF          LD     A,FFH
47 E3A0 D3F6          OUT     (SIOAC),A
48 E3A2 C9           RET
49 E3A3              ;
50 E3A3              ;
51 E3A3              ;
52 E3A3 0EF6          SYNCA: LD     C,F6H      ; C sioac
53 E3A5 ED79          OUT     (C),A
54 E3A7 3E05          LD     A,5
55 E3A9 D3F7          OUT     (SIOBC),A
56 E3AB 3E80          LD     A,80H
57 E3AD D3F7          OUT     (SIOBC),A
58 E3AF EDB3          OTIR
59 E3B1 C9           RET
60 E3B2              ;

```

```

01 E3B2              SKP     H

```

```

01 E3B2      ;
02 E3B2      ;
03 E3B2      ;
04 E3B2      ; EOM -- End off message
05 E3B2      ;       Save CRC#1,#2,FLAG
06 E3B2      ;       File space check
07 E3B2      ;
08 E3B2      ;
09 E3B2      EOM:  ENT      BC,1          ;WAIT 0.7ms
10 E3B2 010100 LD      BC,1
11 E3B5 CD14E4 CALL    TIMW
12 E3B8 3E10  LD      A,10H
13 E3BA D3F7  OUT     (SIOBC),A
14 E3BC DBF7  IN      A,(SIOBC)
15 E3BE E608  AND     8
16 E3C0 C0    RET     NZ
17 E3C1 1846  JR      IOE53          ;NO file space
18 E3C3      ;
19 E3C3      ; RDCRC -- READ CRC & CHECK
20 E3C3      ;
21 E3C3      RDCRC: ENT
22 E3C3 0603  LD      B,3
23 E3C5 CDF0E3 RDCR1: CALL  RDATA
24 E3C8 10FB  DJNZ    RDCR1
25 E3CA DBF6  RDCR2: IN      A,(SIOAC)
26 E3CC 0F    RRCA
27 E3CD 30FB  JR      NC,RDCR2        ; Rx Available
28 E3CF 3E01  LD      A,01H
29 E3D1 D3F6  OUT     (SIOAC),A
30 E3D3 DBF6  IN      A,(SIOAC)
31 E3D5 E640  AND     40H
32 E3D7 2027  JR      NZ,IOE41        ;Hard err
33 E3D9 B7    OR      A
34 E3DA C9    RET
35 E3DB      ;
36 E3DB      SKP     H

```

```

01 E3DB      ;
02 E3DB      ; Save 1 chr by Acc
03 E3DB      ;       & ready check
04 E3DB      ;
05 E3DB      TRANS: ENT
06 E3DB F5    PUSH    AF
07 E3DC DBF6  TRA1: IN      A,(SIOAC)
08 E3DE E604  AND     4          ;TRANS buf null
09 E3E0 28FA  JR      Z,TRA1
10 E3E2 F1    POP     AF
11 E3E3 D3F4  OUT     (SIOAD),A
12 E3E5 3E10  LD      A,10H
13 E3E7 D3F6  OUT     (SIOAC),A
14 E3E9 DBF6  IN      A,(SIOAC)
15 E3EB E608  AND     08H
16 E3ED 2817  JR      Z,IOE50        ;Not ready
17 E3EF C9    RET
18 E3F0      ;
19 E3F0      ; Read data (1 chr)
20 E3F0      ;
21 E3F0      RDATA: ENT
22 E3F0 CDE5E3 CALL    NRCK
23 E3F3 DBF6  IN      A,(SIOAC)        ;RR0
24 E3F5 07    RLCA
25 E3F6 3808  JR      C,IOE41        ;Hard err
26 E3F8 0F    RRCA
27 E3F9 0F    RRCA
28 E3FA 30F4  JR      NC,RDATA
29 E3FC DBF4  IN      A,(SIOAD)
30 E3FE B7    OR      A
31 E3FF C9    RET
32 E400      ;
33 E400      SKP     H

```

```

01 E400      ;
02 E400      ; i/a err
03 E400
04 E400 3E29  IOE41: LD      A,41      ;Hard err
05 E402 21      DEFB    21H
06 E403 3E2E  IOE46: LD      A,46      ;Write protect
07 E405 21      DEFB    21H
08 E406 3E32  IOE50: LD      A,50      ;Not ready
09 E408 21      DEFB    21H
10 E409 3E35  IOE53: LD      A,53      ;No file space
11 E40B 21      DEFB    21H
12 E40C 3E36  IOE54: LD      A,54      ;Un format
13 E40E ED7B4811 LD      SP,(RETSP)
14 E412 37      SCF
15 E413 C9      RET
16 E414      ;
17 E414      ;
18 E414      ;
19 E414      ;
20 E414      ; wait timer
21 E414      ;
22 E414      ; BC=001H= 0.7ms( 0.704ms)
23 E414      ; 003H= 2.0ms( 2.107ms)
24 E414      ; 01DH= 20.0ms( 19.938ms)
25 E414      ; 0A0H=110.0ms(110.050ms)
26 E414      ; 0E9H=160.0ms(160.140ms)
27 E414      ; 140H=220.0ms(219.940ms)
28 E414      ;
29 E414      ;
30 E414      ;
31 E414      TIMW: ENT
32 E414 F5      PUSH AF
33 E415 3E96  TIMW1: LD      A,96H      ; for ex
34 E417 30      TIMW2: DEC     A
35 E418 20FD      JR      NZ,TIMW2
36 E41A 0B      DEC     BC
37 E41B 78      LD      A,B
38 E41C B1      OR      C
39 E41D 20F6      JR      NZ,TIMW1
40 E41F F1      POP     AF
41 E420 C9      RET
42 E421      ;
43 E421      SKP     H

```

```

01 E421      ;
02 E421      ;
03 E421      ;
04 E421      ; SIO CH A COMMAND CHAIN
05 E421      ;
06 E421      ; SIOLD -- LOAD INIT. DATA
07 E421      ; SIOSD -- SAVE INIT. DATA
08 E421      ;
09 E421      ;
10 E421      ;
11 E421 58      SIOLD: DEFB    58H      ;CHANNEL RESET
12 E422 04      DEFB    04H      ;POINT WR4
13 E423 10      DEFB    10H      ;X1 CLOCK
14 E424 05      DEFB    05H      ;POINT WR1
15 E425 04      DEFB    04H      ;CRC-16
16 E426 03      DEFB    03H      ;POINT WR3
17 E427 00      DEFB    00H      ;ENTER HUNT PHASE
18 E428      ;
19 E428 06      ;Rx 8bits DEFB    06H      ;POINT WR6
20 E429 16      DEFB    16H      ;SYNC CHR(1)
21 E42A 07      DEFB    07H      ;POINT WR7
22 E42B 16      DEFB    16H      ;SYNC CHR(2)
23 E42C      ;
24 E42C 98      SIOSD: DEFB    98H      ;CHANNEL RESET
25 E42D      ;Tx CRC Generator reset
26 E42D 04      DEFB    04H      ;POINT WR4
27 E42E 10      DEFB    10H      ;X1 CLOCK
28 E42F 06      DEFB    06H      ;POINT WR6
29 E430 16      DEFB    16H      ;SYNC CHR(1)
30 E431 07      DEFB    07H      ;POINT WR7
31 E432 16      DEFB    16H      ;SYNC CHR(2)
32 E433 05      DEFB    05H      ;POINT WR5
33 E434 6D      DEFB    6DH      ;Tx CRC ENABLE
34 E435      ;
35 E435      ;
36 E435      ; BREAK CHECK
37 E435      ;
38 E435 3EE8      BRKC: LD      A,E8H
39 E437 3200E0      LD      (E000H),A
40 E43A 00      NOP
41 E43B 3A01E0      LD      A,(E001H)
42 E43E E681      AND     81H
43 E440 C0      RET     NZ
44 E441 CDD8E1      CALL    WRCAN
45 E444 ED7B4811 LD      SP,(RETSP)
46 E448 37      SCF
47 E449 C9      RET
48 E44A      ;
49 E44A      SKP     H

```

3-3 Mini-floppy disk control

** Z80 ASSEMBLER Z2-004C <92-504M-V1.0C> PAGE 23

03.06.85

** Z80 ASSEMBLER Z2-004C

```

01 E44A      ;
02 E44A      ;----< MFM Minifloppy control >----
03 E44A      ;      Subroutine (MB8876)
04 E44A      ;
05 E44A      ;
06 E44A      ;
07 E44A      ;      Call condition
08 E44A      ;
09 E44A      ;      Case of disk initialize
10 E44A      ;      DRIVE N=IX+0 (0~3)
11 E44A      ;
12 E44A      ;      Case of sequential read & write
13 E44A      ;      Drive n=IX+0 (0~3)
14 E44A      ;
15 E44A      ;      Sector adrs =IX+1,2 (0-045FH)
16 E44A      ;      (0-1119 )
17 E44A      ;      Byte size =IX+3,4
18 E44A      ;      Address =IX+5,6
19 E44A      ;      Next track =IX+7
20 E44A      ;      Next sector =IX+8
21 E44A      ;      Start track =IX+9
22 E44A      ;      Start sector =IX+10
23 E44A      ;
24 E44A      ;      I/O Port address
25 E44A      ;
26 E44A      ;
27 00D8 P    CR:      EQU      D8H
28 00D9 P    TR:      EQU      D9H
29 00DA P    SCR:     EQU      DAH
30 00DB P    DR:      EQU      DBH
31 00DC P    DM:      EQU      DCH
32 00DD P    HS:      EQU      DDH
33 E44A      ;
34 E44A      ;      Subroutine work
35 E44A      ;
36 CF00 P    BPRO:    EQU      CF00H      ;IFM loading adrs.
37 11A3 P    BUF:     EQU      11A3H
38 CEE9 P    BPARA:   EQU      BPRO-23    ;FD work
39 E44A      ;
40 E44A      ;
41 E44A      ;-----< Ercode map >-----
42 E44A      ;
43 E44A      ; 50 :Not ready
44 E44A      ; 41 :Data error
45 E44A      ;
46 E44A      ;      Track 00 err
47 E44A      ;      Write protect err
48 E44A      ;      Seek err
49 E44A      ;      CRC err
50 E44A      ;      Lost data
51 E44A      ; 54 :Unformat
52 E44A      ;      Recode not found
53 E44A      ; 56 :Invalid data
54 E44A      ;
55 E44A      ;      Main routine
56 E44A      ;
57 E44A      ;
58 E44A E3    FDX:     ENT      (SP),HL
59 E44B 22FECE LD      (BPARA+21),HL
60 E44E C0D5E8 CALL    FDCC&      ;FD i/o check

```

```

01 E451 C2BAE4      JP      NZ,NOTIO
02 E454 11E9CE      LD      DE,BPARA
03 E457 21D1E4      LD      HL,BOOT
04 E45A 010800      LD      BC,11
05 E45D E0B0        LDIR
06 E45F C030E5      CALL    MOFF
07 E462 DD21E9CE    SJP:    LD      IX,BPARA
08 E466 C0A7E5      CALL    BREAD
09 E469 2100CF      LD      HL,BPRO
10 E46C 11CAE4      LD      DE,IPLMC
11 E46F 0607        LD      B,7
12 E471 4E          MCHECK: LD      C,(HL)
13 E472 1A          LD      A,(DE)
14 E473 B9          CP      C
15 E474 C2AEE4      JP      NZ,MASTE
16 E477 23          INC     HL
17 E478 13          INC     DE
18 E479 10F6        DJNZ    MCHECK
19 E47B 1188ED      LD      DE,IPLM0
20 E47E DF          RST     3
21 E47F 1107CF      LD      DE,BPRO+7
22 E482 DF          RST     3
23 E483 210012      LD      HL,1200H
24 E486 DD7505      LD      (IX+5),L
25 E489 DD7406      LD      (IX+6),H
26 E48C 2A14CF      LD      HL,(BPRO+14H)
27 E48F DD7503      LD      (IX+3),L
28 E492 DD7404      LD      (IX+4),H
29 E495 2A1ECF      LD      HL,(BPRO+1EH)
30 E498 DD7501      LD      (IX+1),L
31 E49B DD7402      LD      (IX+2),H
32 E49E             ;
33 E49E C0A7E5      ;      CALL    BREAD
34 E4A1 C030E5      ;      CALL    MOFF
35 E4A4             ;
36 E4A4             ;      Exec load file
37 E4A4             ;
38 E4A4 010002      LD      BC,0200H
39 E4A7 D9          EXX
40 E4A8 2114CF      LD      HL,BPRO+14H
41 E4AB C3FCEC      JP      EXF
42 E4AE             ;
43 E4AE C030E5      MASTE:  CALL    MOFF
44 E4B1 11F6ED      LD      DE,ERRM1
45 E4B4 180C        JR      ERRTR1
46 E4B6             ;
47 E4B6 FE32        ERRTR:  CP      50
48 E4B8 2005        JR      NZ,ERRTR0
49 E4BA 11B5ED      NOTIO:  LD      DE,IPLM3
50 E4BD 1803        JR      ERRTR1
51 E4BF 11E5ED      ERRTR0: LD      DE,ERRM0
52 E4C2 31EE10      ERRTR1: LD      SP,10EEH
53 E4C5 2AFCE6      LD      HL,(BPARA+21)
54 E4C8 E3          EX      (SP),HL
55 E4C9 C9          RET
56 E4CA             ;
57 E4CA             ;      PARAMETER SETTING
58 E4CA             ;
59 E4CA 03          IPLMC:  DEFB    03H
60 E4CB 49504C50    DEFB    'IPLPRO'

```

```

01 E4CF 524F      BOOT:  DEFB  0      ;DRIVE NO.
02 E4D1 00        DEFW  0000H     ;SECTOR ADDR.
03 E4D2 0000      DEFW  0100H     ;IFM BYTE SIZE
04 E4D4 0001      DEFW  BPRO      ;IFM LOADING ADDR.
05 E4D6 00CF      DEFW  0        ;IX+7,8
06 E4D8 0000      DEFW  0        ;IX+9,10
07 E4DA 0000
08 E4DC
09 E4DC           ; READY CHECK
10 E4DC           ;
11 E4DC           ;
12 E4DC 3AF5CE     READY:  ENT      A,(MTFG)
13 E4DF 0F         LD          A,(MTFG)
14 E4E0 D417E5     RRCA
15 E4E3 D07E00     CALL      NC,FDMTON
16 E4E6 F684       LD          A,(IX+0)      ;DRIVE NO GET
17 E4E8 D3DC       OR          84H          ;DRIVE SELECT MOTON
18 E4EA AF         OUT         (DM),A
19 E4EB 32F4CE     XOR          A
20 E4EE CDBFE6     LD          (CMD),A
21 E4F1 210000     CALL      DLY60M
22 E4F4 2B         LD          HL,0
23 E4F5 7C         REDY0:  DEC      HL
24 E4F6 B5         LD          A,H
25 E4F7 2819       OR          L
26 E4F9 DBD8       JR          Z,REDY1
27 E4FB 2F         IN          A,(CR)      ;STATUS GET
28 E4FC 07         CPL
29 E4FD 38F5       RLCA
30 E4FF DD4E00     JR          C,REDY0
31 E502 21F6CE     LD          C,(IX+0)
32 E505 0600       LD          HL,CLBF0
33 E507 09         LD          B,0
34 E508 CB46       ADD         HL,BC
35 E50A 2005       BIT          0,(HL)
36 E50C CD48E5     JR          NZ,REDY2
37 E50F C8C6       CALL      RCLB
38 E511 C9         SET         0,(HL)
39 E512 3E32       REDY2:  RET
40 E514 C390E6     REDY1:  LD          A,50
41 E517           JP          ERJMP
42 E517           ;
43 E517           ; MOTOR ON
44 E517           ;
45 E517           ;
46 E517           ;
47 E517           ;
48 E517 3E80       FDMTON: ENT      A,80H
49 E519 D3DC       LD          A,(DM),A
50 E51B 0610       OUT         (DM),A
51 E51D CDBFE6     LD          B,16
52 E51F 10FB       MTD1:  CALL      DLY60M
53 E522 3E01       LD          MTD1
54 E524 32F5CE     LD          A,1
55 E527 C9         LD          (MTFG),A
56 E528           RET
57 E528           ;
58 E528           ; SEEK TREATMENT
59 E528           ;
60 E528           ;
61 E528           ;
62 E528           ;
63 E528           ;
64 E528           ;
65 E528           ;
66 E528           ;
67 E528           ;
68 E528           ;
69 E528           ;
70 E528           ;
71 E528           ;
72 E528           ;
73 E528           ;
74 E528           ;
75 E528           ;
76 E528           ;
77 E528           ;
78 E528           ;
79 E528           ;
80 E528           ;
81 E528           ;
82 E528           ;
83 E528           ;
84 E528           ;
85 E528           ;
86 E528           ;
87 E528           ;
88 E528           ;
89 E528           ;
90 E528           ;
91 E528           ;
92 E528           ;
93 E528           ;
94 E528           ;
95 E528           ;
96 E528           ;
97 E528           ;
98 E528           ;
99 E528           ;
100 E528          ;

```

```

01 E530           ;
02 E530           ; MOTOR OFF
03 E530           ;
04 E530           ;
05 E530 F5        MOFF:  ENT      PUSH  AF
06 E531 CDB9E6     CALL      DLY1M      ;1000 US DELAY
07 E534 AF        XOR          A
08 E535 D3DC       OUT         (DM),A
09 E537 32F6CE     LD          (CLBF0),A
10 E53A 32F7CE     LD          (CLBF1),A
11 E53D 32F8CE     LD          (CLBF2),A
12 E540 32F9CE     LD          (CLBF3),A
13 E543 32F5CE     LD          (MTFG),A
14 E546 F1        POP          AF
15 E547 C9        RET
16 E548           ;
17 E548           ; RECALIBRATION
18 E548           ;
19 E548           ;
20 E548 3E0B       RCLB:  ENT      A,0BH
21 E54A CD55E5     LD          A,0BH
22 E54D E685       CALL      CMDOT1
23 E54F EE04       AND          85H
24 E551 C8        XOR          04
25 E552 C361E6     RET          Z
26 E555           JP          ERROR
27 E555           ;
28 E555           ; COMAND OUT ROUTINE
29 E555 32F4CE     ;
30 E558 2F        CMDOT1: LD          (CMD),A
31 E559 D3D8       CPL
32 E55B CD68E5     OUT         (CR),A
33 E55E CDBFE6     CALL      BSYON
34 E561 DBD8       CALL      DLY60M
35 E563 2F        IN          A,(CR)
36 E564 32FBCE     CPL
37 E567 C9        LD          (STAFG),A
38 E568           RET
39 E568           ;
40 E568           ; BUSY AND WAIT
41 E568 D5        ;
42 E569 E5        BSYON:  PUSH      DE
43 E56A CDA1E5     PUSH      HL
44 E56D 210000     CALL      BSY0
45 E570 2B        BSYON2: LD          HL,0000H
46 E571 7C        BSYON0: DEC      HL
47 E572 B5        LD          A,H
48 E573 2808       OR          L
49 E575 DBD8       JR          Z,BSYON1
50 E577 0F        IN          A,(CR)
51 E578 30F6       RRCA
52 E57A E1        BSYON3: JR          NC,BSYON0
53 E57B D1        POP          HL
54 E57C C9        POP          DE
55 E57D           RET
56 E57D 10        ;
57 E57E 20ED       BSYON1: DEC      E
58 E580 3E29       JR          NZ,BSYON2
59 E582 E1        BSYONE: LD          A,41
60 E583 01        POP          HL

```

```

01 E584 C390E6      JP      ERJMP
02 E587      ;
03 E587 D5      BSYOFF: PUSH DE
04 E588 E5      PUSH HL
05 E589 CDA1E5      CALL BSY0
06 E58C 210000      BSYOF2: LD HL,0000H
07 E58F 2B      BSYOF0: DEC HL
08 E590 7C      LD A,H
09 E591 B5      OR L
10 E592 2808      JR Z,BSYOF1
11 E594 DBD8      IN A,(CR)
12 E596 0F      RRCA
13 E597 38F6      BSYOF3: JR C,BSYOF0
14 E599 E1      POP HL
15 E59A D1      POP DE
16 E59B C9      RET
17 E59C      ;
18 E59C 1D      BSYOF1: DEC E
19 E59D 20E0      JR NZ,BSYOF2
20 E59F 18DF      BSYOF2: JR BSYONE
21 E5A1      ;
22 E5A1 CDB3E6      BSY0: CALL DLY80U
23 E5A4 1E07      LD E,07H
24 E5A6 C9      RET
25 E5A7      ;
26 E5A7      ;
27 E5A7      ; SEQUENTIAL READ
28 E5A7      ;
29 E5A7      BREAD: ENT
30 E5A7 CD96E6      CALL CNVRT
31 E5AA CDFBE5      READ1: CALL PARST1
32 E5AD CD1BE6      RE8: CALL SIDST
33 E5B0 CD28E5      CALL SEEK
34 E5B3 C290E6      JP NZ,ERJMP
35 E5B6 CD2BE6      CALL PARST2
36 E5B9 F3      DI
37 E5BA 3E94      LD A,94H
38 E5BC CD4EE6      CALL CMDOT2
39 E5BF 0600      LD B,00
40 E5C1 DBD8      RE6: IN A,(CR)
41 E5C3 0F      RE4: RRCA
42 E5C4 381A      JR C,RE3
43 E5C6 0F      RRCA
44 E5C7 38F8      JR C,-6
45 E5C9 EDA2      INI
46 E5CB 20F4      JR NZ,RE4
47 E5CD DD3408      INC A,(IX+8)
48 E5D0 DD7E08      LD A,(IX+8)
49 E5D3 FE11      CP 17
50 E5D5 2805      JR Z,+7
51 E5D7 15      DEC D
52 E5D8 20E5      JR NZ,RE6
53 E5DA 1801      JR +3
54 E5DC 15      DEC D
55 E5DD CD58E6      CALL INTER
56 E5E0 00      RE3: NOP
57 E5E1 DBD8      IN A,(CR)
58 E5E3 2F      CPL
59 E5E4 32FBCE      LD (STAFG),A
60 E5E7 E6FF      AND FFH

```

```

01 E5E9 2076      JR NZ,READY
02 E5EB CD3CE6      CALL AUJ
03 E5EE CAF6E5      JP Z,REND
04 E5F1 DD7E07      LD A,(IX+7)
05 E5F4 18B7      JR RE8
06 E5F6 3E80      REND: LD A,80H
07 E5F8 D3DC      OUT (DM),A
08 E5FA C9      RET
09 E5FB      ;
10 E5FB      ;
11 E5FB      ;
12 E5FB      ; PARAMETER SET
13 E5FB      ;
14 E5FB      ;
15 E5FB CDDCE4      PARST1: CALL READY
16 E5FE DD5604      LD D,(IX+4)
17 E601 DD7E03      LD A,(IX+3)
18 E604 B7      OR A
19 E605 2801      JR Z,+3
20 E607 14      INC D
21 E608 DD7E0A      LD A,(IX+10)
22 E60B DD7708      LD (IX+8),A
23 E60E DD7E09      LD A,(IX+9)
24 E611 DD7707      LD (IX+7),A
25 E614 DD6E05      LD L,(IX+5)
26 E617 DD6606      LD H,(IX+6)
27 E61A C9      RET
28 E61B      ;
29 E61B      ; SIZE SEEK SET
30 E61B      ;
31 E61B CB3F      SIDST: SRL A
32 E61D 2F      CPL
33 E61E D3DB      OUT (DR),A
34 E620 3004      JR NC,+6
35 E622 3E01      LD A,1
36 E624 1801      JR +3
37 E626 AF      XOR A
38 E627 2F      CPL
39 E628 D3DD      OUT (HS),A
40 E62A C9      RET
41 E62B      ;
42 E62B      ;
43 E62B      ; TRACK & SECTOR SET
44 E62B      ;
45 E62B 0EDB      PARST2: LD C,DBH
46 E62D DD7E07      LD A,(IX+7)
47 E630 CB3F      SRL A
48 E632 2F      CPL
49 E633 D3D9      OUT (TR),A
50 E635 DD7E08      LD A,(IX+8)
51 E638 2F      CPL
52 E639 D3DA      OUT (SCR),A
53 E63B C9      RET
54 E63C      ;
55 E63C      ;
56 E63C      ; ADJUST SECT & TRACK
57 E63C      ;
58 E63C DD7E08      ADJ: LD A,(IX+8)
59 E63F FE11      CP 17
60 E641 2008      JR NZ,+10

```

```

01 E643 3E01          LD      A,1
02 E645 0D7708        LD      (IX+8),A
03 E648 0D3407        INC     (IX+7)
04 E64B 7A            LD      A,0
05 E64C B7            OR      A
06 E64D C9            RET
07 E64E                ;
08 E64E                ; COMAND OUT & WAIT
09 E64E                ;
10 E64E 32F4CE        CMDOT2: LD      (CMD),A
11 E651 2F            CPL
12 E652 0308          OUT      (CR),A
13 E654 CD87E5        CALL    BSYOFF
14 E657 C9            RET
15 E658                ;
16 E658                ;
17 E658                ; FORCE INTERRUPT
18 E658                ;
19 E658 3ED8          INTER: LD      A,D8H
20 E65A 2F            CPL
21 E65B D3D8          OUT      (CR),A
22 E65D CD68E5        CALL    BSYON
23 E660 C9            RET
24 E661                ;
25 E661                ;
26 E661                ; STATUS CHECK
27 E661                ;
28 E661 3AF4CE        ERROR: LD      A,(CMD)
29 E664 FE0B          CP      0BH
30 E666 281B          JR      Z,ERCK1
31 E668 FE1B          CP      1BH
32 E66A 2817          JR      Z,ERCK1
33 E66C FEF4          CP      F4H
34 E66E 2813          JR      Z,ERCK1
35 E670 3AFBCE        LD      A,(STAFG)
36 E673 CB7F          BIT      7,A
37 E675 2017          JR      NZ,ERRET
38 E677 CB77          BIT      6,A
39 E679 200F          JR      NZ,ERRET1
40 E67B CB67          BIT      4,A
41 E67D 3E36          LD      A,54
42 E67F 200F          JR      NZ,ERJMP
43 E681 1807          JR      ERRET1
44 E683 3AFBCE        ERCK1: LD      A,(STAFG)
45 E686 CB7F          BIT      7,A
46 E688 2004          JR      NZ,ERRET
47 E68A 3E29          ERRET1: LD      A,41
48 E68C 1802          JR      ERJMP
49 E68E 3E32          ERRET: LD      A,50
50 E690 CD30E5        ERJMP: CALL    MOFF
51 E693 C3B6E4        JP      ERRTRT
52 E696                ;
53 E696                ;
54 E696                ; SECTOR TO TRACK & SECTOR CONVERT
55 E696                ;
56 E696 0600          CNVRT: LD      B,0
57 E698 111000        LD      DE,16
58 E69B 0D6E01        LD      L,(IX+1)
59 E69E 0D6602        LD      H,(IX+2)
60 E6A1 AF            XOR      A

```

```

01 E6A2 ED52          TRANS0: SBC     HL,DE
02 E6A4 3803          JR      C,TRANS1
03 E6A6 04            INC     B
04 E6A7 18F9          JR      TRANS0
05 E6A9 19            TRANS1: ADD     HL,DE
06 E6AA 60            LD      H,B
07 E6AB 2C            INC     L
08 E6AC 0D7409        LD      (IX+9),H
09 E6AF 0D750A        LD      (IX+10),L
10 E6B2 C9            RET
11 E6B3                ;
12 E6B3                ;
13 E6B3                ; TIME DELAY ( 1M & 60M & 80U )
14 E6B3                ;
15 E6B3                ;
16 E6B3 D5            DLY80U: ENT     DE
17 E6B4 110F00        PUSH    DE,15
18 E6B7 180A          JR      DLYT
19 E6B9                ;
20 E6B9 D5            DLY1M: ENT     DE
21 E6BA 11A000        PUSH    DE,160
22 E6BD 1804          JR      DLYT
23 E6BF                ;
24 E6BF D5            DLY60M: ENT     DE
25 E6C0 112620        PUSH    DE,8230
26 E6C3 1B            LD      DE,8230
27 E6C4 7B            DLYT: DEC     DE
28 E6C5 B2            LD      A,E
29 E6C6 20FB          OR      D
30 E6C8 01            JR      NZ,DLYT
31 E6C9 C9            POP     DE
32 E6CA                ;
33 E6CA                ; FLOPPY WORK AREA
34 E6CA                ;
35 E6CA P             CMD: EQU      BPARA+11
36 E6CA P             MTFG: EQU     CMD+1
37 E6CA P             CLBF0: EQU    MTFG+1
38 E6CA P             CLBF1: EQU    CLBF0+1
39 E6CA P             CLBF2: EQU    CLBF1+1
40 E6CA P             CLBF3: EQU    CLBF2+1
41 E6CA P             VRFcnt: EQU    CLBF3+1
42 E6CA P             STAFG: EQU    VRFcnt+1
43 E6CA                ;
44 E6CA                ; SKP      H

```

```

01 E6CA ;
02 E6CA ;=====
03 E6CA ;
04 E6CA ;      1R12(Static ram board 32Kbyte)
05 E6CA ;      Read/Write routine
06 E6CA ;
07 E6CA ;=====
08 E6CA ;
09 1108 P COMNT: EQU 1108H
10 E6CA ;
11 E6CA ; 1R12 Read sub.
12 E6CA ;
13 E6CA 1R12RD: ENT
14 E6CA CALL 1R12R0
15 E6CD JP NZ,?ERX
16 E6D0 ;
17 E6D0 ; Exec load file
18 E6D0 ;
19 E6D0 EXRAM: ENT
20 E6D0 LD BC,0000H ; Default code (SRAM)
21 E6D3 EXX
22 E6D4 LD HL,SIZE
23 E6D7 JP EXF
24 E6DA ;
25 E6DA 1R12R0: ENT
26 E6DA ;
27 E6DA ; Information sum check
28 E6DA ;
29 E6DA CALL CHECK
30 E6DD RET NZ
31 E6DE ;
32 E6DE ; Counter reset
33 E6DE ;
34 E6DE ; IN A,(C)
35 E6E0 ;
36 E6E0 ; Read information block (9Byte)
37 E6E0 ;
38 E6E0 INC C
39 E6E1 LD HL,SIZE
40 E6E4 LD B,09H
41 E6E6 INIR
42 E6E8 ;
43 E6E8 ; Read data block
44 E6E8 ;
45 E6E8 LD DE,(SIZE)
46 E6EC LD HL,1200H ; Loading adrs
47 E6EF LD A,E
48 E6F0 OR A
49 E6F1 JR Z,+5
50 E6F3 LD B,A
51 E6F4 INIR
52 E6F6 LD B,00H
53 E6F8 DEC D
54 E6F9 JP P,-5
55 E6FC ;
56 E6FC ; Data block's sum check
57 E6FC ;
58 E6FC LD DE,1200H ; Data adrs
59 E6FF LD BC,(SIZE) ; BCSIZE
60 E703 CALL SUM ; HLCheck sum

```

```

01 E706 ED5B0811 LD DE,(COMN1) ; DECheck sum datq
02 E70A B7 OR A
03 E70B ED52 SBC HL,DE ; OK?
04 E70D C9 RET
05 E70E ;
06 E70E ; sum check
07 E70E ;
08 E70E ; IN BC=Size
09 E70E ; DE=Data adrs
10 E70E ; EXIT HL=Check sum
11 E70E ;
12 E70E D9 SUM: EXX
13 E70F 210000 LD HL,0 ; HL'= Check sum clr
14 E712 0E08 LD C,8 ; C' = Loop count
15 E714 D9 EXX
16 E715 78 SUMCK1: LD A,B ; BC = Size
17 E716 B1 OR C
18 E717 280E JR Z,SUMCK2 ; DE = Data adrs
19 E719 1A LD A,(DE)
20 E71A D9 EXX
21 E71B 41 LD B,C ; BC'
22 E71C 07 SUMCK3: RLCA
23 E71D 3001 JR NC,+3
24 E71F 23 INC HL ; HL'= Check sum data
25 E720 10FA DJNZ SUMCK3
26 E722 D9 EXX
27 E723 13 INC DE ; DE
28 E724 0B DEC BC ; BC
29 E725 18EE JR SUMCK1
30 E727 D9 SUMCK2: EXX
31 E728 C9 RET
32 E729 ;
33 E729 ; Information's sum check
34 E729 ;
35 E729 ED78 CHECK: IN A,(C) ; Counter reset
36 E72B 0608 LD B,08H ; B=Byte Counter
37 E72D 1600 LD D,00H ; Sum Counter
38 E72F 0C INC C ; C=C+1
39 E730 ED78 CK1: IN A,(C) ; C=C+1
40 E732 C5 PUSH BC
41 E733 0608 LD B,08H ; Bit Counter
42 E735 07 CK2: RLCA
43 E736 3001 JR NC,+3
44 E738 14 INC D
45 E739 10FA DJNZ CK2
46 E73B 7A LD A,D
47 E73C C1 POP BC
48 E73D 57 LD D,A
49 E73E 10F0 DJNZ CK1
50 E740 ED78 IN A,(C) ; C=C+1
51 E742 0D DEC C ; C=C+0
52 E743 BA CP D
53 E744 C9 RET
54 E745 ;
55 E745 ;=====
56 E745 ;
57 E745 ; 1R12 Write sub.
58 E745 ;
59 E745 ;=====
60 E745 ;

```

```

01 E745      1R12WR: ENT
02 E745      ;
03 E745      ;   Disp 'Writing.....'
04 E745      ;
05 E745      LD      DE,MSGW01      ; 'Writing now'
06 E748 CD1809 CALL    0918H
07 E748 DF      RST      3
08 E74C CD1809 CALL    0918H
09 E74F 11F0E7 LD      DE,MSGW02      ; 'Master tape set'
10 E752 DF      RST      3
11 E753 CD1809 CALL    0918H
12 E756      ;
13 E756      ;   Read information block
14 E756      ;
15 E756 CD2700 CALL    RDINF
16 E759 380E      JR      C,RDB0
17 E75B      ;
18 E75B      ;   Disp 'Loading.....'
19 E75B      ;
20 E75B      LD      DE,MSGGLD      ; 'Loading'
21 E75E CD1809 CALL    0918H
22 E761 DF      RST      3
23 E762 11F110 LD      DE,NAME
24 E765 DF      RST      3
25 E766      ;
26 E766      ;   Read data block
27 E766      ;
28 E766 CD2A00 CALL    RDDAT
29 E769 DA24EB RDB0: JP      C,?ERX
30 E76C      ;
31 E76C      ;   Counter reset
32 E76C      ;
33 E76C ED78      IN      A,(C)      ; C=C+0
34 E76E      ;
35 E76E      ;   Sum check for data
36 E76E      ;
37 E76E ED430A11 LD      (COMNT+2),BC      ; C=Port adrs,C=C+0
38 E772 ED5B0411 LD      DE,(DTADR)
39 E776 ED4B0211 LD      BC,(SIZE)
40 E77A D5      PUSH    DE
41 E77B C5      PUSH    BC
42 E77C CD0EE7 CALL    SUM      ; HLCHECK sum
43 E77F 220811 LD      (COMNT),HL
44 E782      ;
45 E782      ;   Write information (8Byte)
46 E782      ;
47 E782 210211 LD      HL,SIZE
48 E785 ED4B0A11 LD      BC,(COMNT+2)      ; C=Port adrs,C=C+0
49 E789 0608      LD      B,08H      ; Byte Counter
50 E78B 0C      INC      C
51 E78C 0C      INC      C
52 E78D E5      PUSH    HL
53 E78E C5      PUSH    BC
54 E78F EDB3      OTIR      ; C=C+2
55 E791 C1      POP      BC
56 E792 E1      POP      HL
57 E793      ;
58 E793      ;   Sum check for information block
59 E793      ;   AccCheck sum data
60 E793      ;

```

```

01 E793 D5      PUSH    DE      DESize
02 E794 1600      LD      D,00H      Sum Counter
03 E796 C5      WCK1: PUSH    BC
04 E797 0608      LD      B,08H
05 E799 7E      LD      A,(HL)
06 E79A 07      WCK2: RLCA
07 E79B 3001      JR      NC,WCK3
08 E79D 14      INC      D
09 E79E 10FA      WCK3: DJNZ   WCK2
10 E7A0 23      INC      HL
11 E7A1 C1      POP      BC
12 E7A2 10F2      DJNZ   WCK1
13 E7A4 7A      LD      A,D
14 E7A5 D1      POP      DE
15 E7A6 ED79      OUT     (C),A      ; C=C+2
16 E7A8      ;
17 E7A8      ;   Write data block
18 E7A8      ;
19 E7A8 D1      POP      DE      ; DESize
20 E7A9 E1      POP      HL      ; HLData adrs
21 E7AA 7B      LD      A,E
22 E7AB B7      OR      A
23 E7AC 2803      JR      Z,+5
24 E7AE 43      LD      B,E
25 E7AF EDB3      OTIR      ; C=C+2
26 E7B1 0600      LD      B,00
27 E7B3 15      DEC      D
28 E7B4 F2AFE7      JP      P,-5
29 E7B7 C35EEA      JP      ST1X
30 E7BA      ;
31 E7BA      ;   1R12 set check
32 E7BA      ;
33 E7BA      ;   R12CK1: ENT
34 E7BA AF      XOR      A      ; carry flag reset
35 E7BB ED78      IN      A,(C)      ; Counter reset,C=C+0
36 E7BD 0C      INC      C
37 E7BE ED78      IN      A,(C)      ; C=C+1
38 E7C0 08      EX      AF,AF'
39 E7C1 0D      DEC      C
40 E7C2 ED78      IN      A,(C)      ; Counter reset
41 E7C4 06A5      LD      B,A5H
42 E7C6 0C      INC      C
43 E7C7 0C      INC      C
44 E7C8 ED41      OUT     (C),B      ; C=C+2
45 E7CA 0D      DEC      C
46 E7CB 0D      DEC      C
47 E7CC ED78      IN      A,(C)      ; Counter reset
48 E7CE 0C      INC      C
49 E7CF ED78      IN      A,(C)      ; C=C+1
50 E7D1 B8      CP      B
51 E7D2 200B      JR      NZ,12NSET
52 E7D4 0D      DEC      C
53 E7D5 ED78      IN      A,(C)      ; Counter reset
54 E7D7 08      EX      AF,AF'
55 E7D8 0C      INC      C
56 E7D9 0C      INC      C
57 E7DA ED79      OUT     (C),A      ; C=C+2
58 E7DC 0D      DEC      C
59 E7DD 0D      DEC      C
60 E7DE C9      RET

```

3-4 ROM Monitor

** Z80 ASSEMBLER 2Z-004C <9Z-504M-V1.0C> PAGE 35

03.06.85

```

01 E7DF AF      12NSET: XOR    A      ; Z flag set
02 E7E0 37      SCF              ; Not set
03 E7E1 C9      RET
04 E7E2         ;
05 E7E2         ;
06 E7E2         ;      Message table
07 E7E2         ;
08 E7E2 52      MSGW01: DEFM    'R'    ; 'Ramcard prog.'
09 E7E3 A1      DEFB    A1H          ; A
10 E7E4 B3      DEFB    B3H          ; M
11 E7E5 9F      DEFB    9FH          ; C
12 E7E6 A1      DEFB    A1H          ; A
13 E7E7 90      DEFB    9DH          ; R
14 E7E8 9C      DEFB    9CH          ; D
15 E7E9 20      DEFB    20H          ;
16 E7EA 9E      DEFB    9EH          ; P
17 E7EB 90      DEFB    9DH          ; R
18 E7EC B7      DEFB    B7H          ; O
19 E7ED 97      DEFB    97H          ; G
20 E7EE 2E      DEFB    2EH          ;
21 E7EF 0D      DEFB    0DH          ;
22 E7F0         ;
23 E7F0 40      MSGW02: DEFM    'M'    ; 'Master tape set'
24 E7F1 A1      DEFB    A1H          ; A
25 E7F2 A4      DEFB    A4H          ; S
26 E7F3 96      DEFB    96H          ; T
27 E7F4 92      DEFB    92H          ; E
28 E7F5 90      DEFB    9DH          ; R
29 E7F6 20      DEFB    20H          ;
30 E7F7 96      DEFB    96H          ; T
31 E7F8 A1      DEFB    A1H          ; A
32 E7F9 9E      DEFB    9EH          ; P
33 E7FA 92      DEFB    92H          ; E
34 E7FB 20      DEFB    20H          ;
35 E7FC A4      DEFB    A4H          ; S
36 E7FD 92      DEFB    92H          ; E
37 E7FE 96      DEFB    96H          ; T
38 E7FF 0D      DEFB    0DH          ;
39 E800         ;
40 E800         ;      SKP    H

```

** Z80 ASSEMBLER 2Z-004C <9Z-504M-V1.0C> PAGE 36

03.06.85

```

01 E800         ORG    E800H
02 E800         ;
03 E800         ;=====
04 E800         ;
05 E800         ;
06 E800         ;      MZ-800
07 E800         ;
08 E800         ;      Rom Monitor & Initial program loader
09 E800         ;
10 E800         ;      V1.0C   1984.10.11
11 E800         ;
12 E800         ;=====
13 E800         ;
14 E800         ;
15 E800         ;
16 11A3 P      BUFR: EQU    11A3H      ; Key buffer
17 00CE P      GDGMD: EQU    00CEH      ; CRTC Mode reg.
18 00CD P      GDGRF: EQU    00CDH      ;      Read format reg.
19 00CC P      GDGWF: EQU    00CCH      ;      Write format reg.
20 E800         ;
21 E800         ;      Jump table
22 E800         ;
23 E800         ;      BEGIN0: ENT      ; MZ-800 Monitor start
24 E800 00      NOP
25 E801 C313E8   JP      BEGIN
26 E804         ;      BGIN00: ENT      ; Monitor command
27 E804 C35EEA   JP      ST1X
28 E807         ;      LCM12: ENT      ; CMT Load $1200~
29 E807 C370EB   JP      LDCMT
30 E80A         ;      SCM12: ENT      ; CMT Save $1200~
31 E80A C3E8EB   JP      SACMT
32 E80D         ;      VCM12: ENT      ; CMT Verify $1200~
33 E80D C309EC   JP      VFCMT
34 E810         ;      QDIO: ENT      ; Quick disk iocs
35 E810 C310E0   JP      QDIOS
36 E813         ;
37 E813         ;
38 E813         ;      BEGIN: ENT      ; 800 Monitor start
39 E813 F3      DI
40 E814 ED56     IM      1             ; Interrupt mode 1
41 E816 3E08     LD      A,08H         ; '0000 1000'
42 E818 D3CE     OUT     (GDGMD),A     ; 700 Mode
43 E81A 3E01     LD      A,01H         ; '0000 0001'
44 E81C D3CD     OUT     (GDGRF),A     ; Read format reg.
45 E81E D3CC     OUT     (GDGWF),A     ; Write format reg.
46 E820 D3E4     OUT     (E4H),A       ; Initial Bank
47 E822 31F010   LD      SP,SP         ; Stack set (10F0H)
48 E825 CD3E07   CALL    ?MODE         ; 8255 MODE SET
49 E828 AF       XOR      A
50 E829 110000   LD      DE,0000H
51 E82C CD0803   CALL    ?TMST         ; 8253 Mode set
52 E82F         ;
53 E82F         ;      Wait (25.342ms) for Printer
54 E82F         ;
55 E82F 012500   LD      BC,25H
56 E832 CD14E4   CALL    TIMW
57 E835         ;
58 E835         ;      Z80 PIO Initialize
59 E835         ;
60 E835 01FC04   PIOA: LD      BC,04FCH ; Port=A (C;port B;cont)

```

```

01 E838 219AEE      LD      HL,PIOIDA      ; Pio initialize data
02 E838 EDB3        OTIR
03 E83D 01FD04      LD      BC,04FDH      ; Port=B (C;port B;cont)
04 E840 EDB3        OTIR
05 E842            ;
06 E842            ; SIO Initialize
07 E842            ;
08 E842 3E01        LD      A,01H
09 E844 D3F7        OUT     (SIOBC),A
10 E846 AF          XOR     A
11 E847 D3F7        OUT     (SIOBC),A      ; WRI Reg. Clear
12 E849            ;
13 E849            ; (CTRL+Reset) Check
14 E849            ;
15 E849 CD320A      CALL    ?BRK          ; CTRL ?
16 E84C 3014        JR      NC,PSG0
17 E84E FE20        CP      20H          ; Key is 'CTRL' Key
18 E850 C262E8      JP      NZ,PSG0
19 E853 DBCE        IN      A,(CEH)      ; Mode switch
20 E855 E602        AND     2           ; bit 1 check
21 E857 2806        JR      Z,CMY00      ; 700 Mode
22 E859 AF          XOR     A
23 E85A D3CE        OUT     (CEH),A      ; 800 Mode
24 E85C CDE1E8      CALL    PLTST        ; Pallet set
25 E85F C35B00      CMY00: JP      CMY0   ; Bank & Jump
26 E862            ;
27 E862            ; PSG Reset
28 E862            ;
29 E862 0604        LD      B,04H
30 E864 3E9F        LD      A,9FH
31 E866 D3F2        OUT     (F2H),A
32 E868 C620        ADD     A,20H
33 E86A 10FA        DJNZ    PSGSET
34 E86C            ;
35 E86C            ; Sound & Interrupt mask reset
36 E86C            ;
37 E86C 3E01        LD      A,01H
38 E86E 3203E0      LD      (E003H),A    ; Sound mask reset
39 E871 3E05        LD      A,05H
40 E873 3203E0      LD      (E003H),A    ; Interrupt mask reset
41 E876            ;
42 E876            ;
43 E876            ; BGI0:
44 E876 06FF        ENT     B,FFH        ; Buffer Clear
45 E878 21F110      LD      HL,NAME      ; 10F1-11F0H Clear
46 E87B CDD80F      CALL    ?CLER
47 E87E 3E16        LD      A,16H
48 E880 CD1200      CALL    PRNT
49 E883 3E71        LD      A,71H
50 E885 2100D8      LD      HL,D800H
51 E888 CDD509      CALL    #CLR8
52 E88B 218D03      LD      HL,TIMIN
53 E88E 3EC3        LD      A,C3H
54 E890 323810      LD      (1038H),A
55 E893 223910      LD      (1039H),HL
56 E896 3E04        LD      A,04H
57 E898 329E11      LD      (TEMPW),A
58 E89B CDBE02      CALL    MLDSP
59 E89E CDD900      CALL    NL
60 E8A1 C07705      CALL    ?BEL

```

```

01 E8A4 3E01        LD      A,01H      ; Key in silent
02 E8A6 329D11      LD      (SWRK),A
03 E8A9            ;
04 E8A9            ; PCG CG ROM
05 E8A9            ;
06 E8A9 DBE0        IN      A,(E0H)      ; CG & Vram
07 E8AB 1100C0      LD      DE,C000H    ; Destination
08 E8AE 210010      LD      HL,1000H    ; Source
09 E8B1 010010      LD      BC,1000H    ; Size
10 E8B4 EDB0        LDIR
11 E8B6 DBE1        IN      A,(E1H)      ; 1000~1FFFFH,C000~CFFFH
12 E8B8            ;
13 E8B8            ; Key check
14 E8B8            ;
15 E8B8 C01B00      KEYCK: CALL    GETKY      ;Key check
16 E8BB FE40        CP      'M'
17 E8BD 287C        JR      Z,MON        ; Monitor cmd.
18 E8BF FE51        CP      'Q'
19 E8C1 2858        JR      Z,KSJPQB     ; Quick disk
20 E8C3 FE43        CP      'C'
21 E8C5 287E        JR      Z,CSS        ; Cassette tape
22 E8C7            ;
23 E8C7            ; Floppy check & boot
24 E8C7            ;
25 E8C7 CDD5E8      CALL    FDCC&        ; FD i/o check
26 E8CA 2026        JR      NC,R12C&B
27 E8CC CD59EA      FDB:  CALL    DISCLR
28 E8CF CD4AE4      CALL    FDX
29 E8D2 C334EA      JP      ERRMG
30 E8D5            ;
31 E8D5 3EA5        ENT     A,A5H
32 E8D7 47          LD      B,A
33 E8D8 D3D9        OUT     (D9H),A
34 E8DA CDB3E6      CALL    DLY80U
35 E8DD DBD9        IN      A,(D9H)
36 E8DF B8          CP      B
37 E8E0 C9          RET
38 E8E1            ;
39 E8E1            ;
40 E8E1            ; Pallet Reg. & Border Reg. set
41 E8E1            ;
42 E8E1            ; PLT0~3 Black
43 E8E1            ; Border Black
44 E8E1            ;
45 E8E1            ;
46 E8E1 E5          PLTST: ENT     HL
47 E8E2 01F005      PUSH    HL
48 E8E8 EDB3        LD      BC,05F0H    ; C=port , B=counts
49 E8EA AF          LD      HL,PLTDT    ; Data
50 E8EB 01CF06      OTIR
51 E8EE ED79        XOR     A
52 E8F0 E1          LD      BC,06CFH    ; Border Black
53 E8F1 C9          OUT     (C),A
54 E8F2            ; POP     HL
55 E8F2            ; RET
56 E8F2            ;
57 E8F2 0EF8        ; MZ-1R12 check & boot
58 E8F4 CDBAE7      LD      C,F8H
59 E8F7 DAB7E9      CALL    R12CK1      ; Set...cy='0',z='1'
60 E8FA CDDAE6      JP      C,0BT

```

```

01 E8FD 113AEE      LD     DE,12ERMG      ; 'SRAM:Check sum err'
02 E900 C234EA      JP     NZ,ERRMG
03 E903 C3D0E6      JP     EXRAM
04 E906             ;
05 E906             ; Device selection
06 E906             ;
07 E906 CD43EA      SLMSG: CALL  CLLET
08 E909 CD06EA      SELMG0: CALL SELMSG      ; Selection message
09 E90C CD91F1      CALL  NKYWT      ; Wait non key
10 E90F CD13EB      KSCAN0: CALL  SIOCK      ; Sio hard check
11 E912 200A        JR     NZ,KSCAN1
12 E914 3E02        LD     A,02H
13 E916 CD15F1      CALL  SRCK0      ; Key scan
14 E919 FE7F        CP     7FH      ; 'Q'
15 E91B CAB7E9      KSJPQB: JP     Z,QBT
16 E91E 3E04        KSCAN1: LD     A,04H
17 E920 CD15F1      CALL  SRCK0
18 E923 FEDF        CP     DFH      ; 'C'
19 E925 281E        JR     Z,CSS
20 E927 4F          LD     C,A
21 E928 CDD5E8      CALL  FDCC&      ; Fdc check
22 E92B 2005        JR     NZ,KSCAN2
23 E92D 79          LD     A,C
24 E92E FEFB        CP     FBH      ; 'F'
25 E930 289A        JR     Z,FDB
26 E932 3E03        KSCAN2: LD     A,03H
27 E934 CD15F1      CALL  SRCK0
28 E937 FEF7        CP     F7H      ; 'M'
29 E939 20D4        JR     NZ,KSCAN0
30 E93B             ;
31 E93B             ; Monitor jump
32 E93B             ;
33 E93B CD59EA      MON:  CALL  DISCLR
34 E93E 114DEE      LD     DE,MONMSG      ; *** MONITOR ***
35 E941 DF          RST     3
36 E942 C35EEA      JP     ST1X
37 E945             ;
38 E945             ; =====
39 E945             ;
40 E945             ; Cassette boot-up
41 E945             ;
42 E945             ; =====
43 E945             ;
44 E945 CSS:  ENT
45 E945             ;
46 E945             ; Motor check
47 E945             ;
48 E945 2102E0      LD     HL,E002H
49 E948 7E          WMOD0: LD     A,(HL)
50 E949 E610        AND    10H
51 E94B 2025        JR     NZ,DSILF
52 E94D 23          INC     HL      ; PC3:"L""H"
53 E94E 3E06        LD     A,06H
54 E950 77          LD     (HL),A
55 E951 3C          INC     A      ; Acc07
56 E952 77          LD     (HL),A
57 E953 2B          DEC     HL
58 E954 7E          LD     A,(HL)
59 E955 E610        AND    10H
60 E957 2019        JR     NZ,DSILF

```

```

01 E959             ;
02 E959             ; Disp 'Make ready CMT'
03 E959             ;
04 E959 CD59EA      CALL  DISCLR
05 E95C CD0600      CALL  LETNL
06 E95F CD0600      CALL  LETNL
07 E962 1198ED      LD     DE,IPLM1      ; 'Make ready CMT'
08 E965 CD4EEA      CALL  MSGLET
09 E968             ;
10 E968             ; Wait motor on
11 E968             ;
12 E968 CD1E00      WMOD1: CALL  BRKEY      ; Break?
13 E96B 283F        JR     Z,TPERX0
14 E96D 7E          LD     A,(HL)
15 E96E E610        AND    10H
16 E970 28F6        JR     Z,WMOD1
17 E972             ;
18 E972             ; Disp 'IPL is looking for'
19 E972             ;
20 E972 CD59EA      DSILF: CALL  DISCLR
21 E975 CD0600      CALL  LETNL
22 E978 11C3ED      LD     DE,IPLM4      ; 'IPL is looking ...'
23 E97B DF          RST     3
24 E97C             ;
25 E97C             ; Read information block
26 E97C             ;
27 E97C CD2700      CALL  RDINF
28 E97F DAAAE9      JP     C,TPERX
29 E982             ;
30 E982             ; Disp 'IPL is loading...'
31 E982             ;
32 E982 CD59EA      CALL  DISCLR
33 E985 1188ED      LD     DE,IPLM0      ; 'IPL is loading '
34 E988 DF          RST     3
35 E989 11F110      LD     DE,NAME
36 E98C DF          RST     3
37 E98D             ;
38 E98D             ; Load adrs set
39 E98D             ;
40 E98D 2A0411      LD     HL,(DTADR)
41 E990 D9          EXX
42 E991 210012      LD     HL,1200H
43 E994 220411      LD     (DTADR),HL
44 E997             ;
45 E997             ; Read data block
46 E997             ;
47 E997 CD2A00      RDTB0: CALL  RDDAT
48 E99A DAAAE9      JP     C,TPERX
49 E99D             ;
50 E99D             ; Exec load file
51 E99D             ;
52 E99D 010001      EXCMT: LD     BC,0100H      ; Default code (CMT)
53 E9A0 D9          EXX
54 E9A1 220411      LD     (DTADR),HL
55 E9A4 210211      LD     HL,SIZE
56 E9A7 C3FCEC      JP     EXF
57 E9AA             ;
58 E9AA             ; Tape error
59 E9AA             ;
60 E9AA FE02        TPERX: CP     02H      ; BREAK?

```

```

01 E9AC 1198ED      TPERX0: LD    DE,IPLM1      ; 'Make ready CMT'
02 E9AF 2803        JR      2,TPERJP
03 E9B1 1104EE      LD      DE,ERRM2      ; 'CMT:Loading error'
04 E9B4 C334EA      TPERJP: JP      ERRMG
05 E9B7             ;
06 E9B7             ;=====
07 E9B7             ;
08 E9B7             ; Quick disk boot-up
09 E9B7             ;=====
10 E9B7             ;
11 E9B7             ;
12 E9B7             ;
13 E9B7 CD13EB      QBT:  ENT
14 E9BA 3E02        CALL  SIOCK      ; Sio hard check
15 E9BC 20EC        LD      A,02H      ; Break code
16 E9BE CDECEE      JR      NZ,TPERX
17 E9C1 CD27EF      CALL  IOFRS      ; Iocs flag reset
18 E9C4 11A7ED      CALL  QDRCK      ; Media ready check
19 E9C7 386B        LD      DE,IPLM2      ; 'Make ready QD'
20 E9C9 CD59EA      JR      C,ERRMG
21 E9CC 3E0D        CALL  DISCLR      ; Head point clear
22 E9CE 32A311      LD      A,0DH
23 E9D1 CD5FF2      LD      (BUFER),A
24 E9D4             CALL  HDPCL
25 E9D4             ;
26 E9D4             ; Error return set
27 E9D4 3E01        LD      A,01H
28 E9D6 323A11      LD      (QDCPA),A
29 E9D9 2104EA      LD      HL,QDLDER
30 E9DC 31EE10      LD      SP,10EEH
31 E9DF E3          EX      (SP),HL
32 E9E0             ;
33 E9E0             ;
34 E9E0 C0F7EE      QBFLSH: CALL  FILSCH      ; File search
35 E9E3 DA02F2      JP      C,QERTRT
36 E9E6 3AF010      LD      A,(ATRB)
37 E9E9 FE01        CP      OBJCD
38 E9EB 1127EE      LD      DE,ERRM4      ; 'QD:File made error'
39 E9EE 200A        JR      NZ,QDFMER
40 E9F0             ;
41 E9F0             ; Quick disk boot
42 E9F0             ;
43 E9F0 1188ED      LD      DE,IPLM0      ; 'Ipl is loading'
44 E9F3 DF          RST      3
45 E9F4 C3C2EE      JP      DSFLNA
46 E9F7 11A7ED      QDNTR: LD      DE,IPLM2      ; 'Make ready QD'
47 E9FA D5          QDFMER: PUSH  DE
48 E9FB 3E06        LD      A,06H      ; Motor off
49 E9FD 323011      LD      (QDPA),A
50 EA00 CD10E0      CALL  QDIOS
51 EA03 D1          POP      DE
52 EA04 182E        QDLDER: JR      ERRMG
53 EA06             ;
54 EA06             ;=====
55 EA06             ;
56 EA06             ; Disp message
57 EA06             ;=====
58 EA06             ;
59 EA06             ;
60 EA06             ;

```

```

01 EA06             ; Selection message
02 EA06             ;
03 EA06 CD0600      SELMSG: CALL  LETNL
04 EA09 1143ED      LD      DE,SELM0      ; 'Please push key'
05 EA0C CD4EEA      CALL  MSGLET
06 EA0F CD0600      CALL  LETNL
07 EA12 CD05E8      CALL  FDCC8      ; Fdc check
08 EA15 2006        JR      NZ,SELM00
09 EA17 1153ED      LD      DE,SELM1      ; 'F:Flappy disk'
10 EA1A CD4EEA      CALL  MSGLET
11 EA1D CD13EB      SELM00: CALL  SIOCK      ; Sio check
12 EA20 2006        JR      NZ,SELM01
13 EA22 1161ED      LD      DE,SELM2      ; 'Q:Quick disk'
14 EA25 CD4EEA      CALL  MSGLET
15 EA28 116EED      SELM01: LD      DE,SELM3      ; 'C:Cassette tape'
16 EA2B CD4EEA      CALL  MSGLET
17 EA2E 117EED      LD      DE,SELM4      ; 'M:Rom monitor'
18 EA31 C30000      JP      MSGLET
19 EA34             ;
20 EA34             ;
21 EA34             ; Error message
22 EA34             ;
23 EA34 CD59EA      ERRMG: CALL  DISCLR
24 EA37 CD0600      CALL  LETNL
25 EA3A CD0600      CALL  LETNL
26 EA3D CD4EEA      CALL  MSGLET
27 EA40 C309E9      JP      SELMG0
28 EA43             ;
29 EA43             ; Disp clear 3-let
30 EA43             ;
31 EA43 CD59EA      CLLET: CALL  DISCLR
32 EA46 0603        LD      B,03H
33 EA48 CD0600      LET:  CALL  LETNL
34 EA4B 10FB        DJNZ  LET
35 EA4D C9          RET
36 EA4E             ;
37 EA4E             ; Letnl-message
38 EA4E             ;
39 EA4E 060C        MSGLET: LD      B,0CH
40 EA50 CD0C00      MSGPS: CALL  PRNPS
41 EA53 10FB        DJNZ  MSGPS
42 EA55 DF          RST      3
43 EA56 C30000      JP      LETNL
44 EA59             ;
45 EA59             ; Disp clear
46 EA59             ;
47 EA59             ;
48 EA59 3EC6        DISCLR: LD      A,C6H
49 EA5B C3DC0D      JP      ?DPCT
50 EA5E             ;
51 EA5E             ;=====
52 EA5E             ;
53 EA5E             ; Monitor command
54 EA5E             ;=====
55 EA5E             ;
56 EA5E             ;
57 EA5E             ; Get command routine
58 EA5E             ;
59 EA5E             ;
60 EA5E             ; STIX:  ENT

```

```

01 EASE 31F010      LD SP,SP
02 EA61 C00900      CALL NL
03 EA64 3E2A        LD A,2AH      ; "*"
04 EA66 C01200      CALL PRNT
05 EA69 11A311      LD DE,BUFER
06 EA6C C00300      CALL GETL
07 EA6F 1A          LD A,(DE)
08 EA70 13          INC DE
09 EA71 FE0D        CP 0DH
10 EA73 28E9        JR Z,ST1X
11 EA75 FE4A        CP 'J'      ; JUMP
12 EA77 2830        JR Z,GOTOX
13 EA79 FE4C        CP 'L'      ; Load cmt
14 EA7B CA4CEB      JP Z,LOADX
15 EA7E FE46        CP 'F'      ; Floppy boot
16 EA80 287E        JR Z,FDCK
17 EA82 FE42        CP 'B'      ; Bell
18 EA84 CA1EEC      JP Z,SGX
19 EA87 FE4D        CP 'M'      ; Memory correction
20 EA89 CA7BEB      JP Z,MCORX
21 EA8C FE53        CP 'S'      ; Save CMT
22 EA8E CAAEEB      JP Z,SAVEX
23 EA91 FE56        CP 'V'      ; Verify
24 EA93 CA00EC      JP Z,VRFYX
25 EA96 FE44        CP 'D'      ; Dump memory
26 EA98 CA29EC      JP Z,DUMPX
27 EA9B FE51        CP 'Q'      ; Quick disk cmd.
28 EA9D 2816        JR Z,QUICK
29 EA9F FE45        CP 'E'      ; Exit rambrd(1R12) cmd.
30 EAA1 2842        JR Z,RAMBRD
31 EAA3 FE47        CP 'G'      ; Call cmd.
32 EAA5 2806        JR Z,CAL0
33 EAA7 18C6        JR ST2X
34 EAA9
35 EAA9             ;
36 EAA9             ;      Jump command
37 EAA9 C042EB      GOTOX: CALL HEXIYX
38 EAAE E9          GOTO: JP (HL)
39 EAAD
40 EAAD             ;
41 EAAD             ;      Call command
42 EAAD C042EB      CAL0: CALL HEXIYX
43 EAB0 CDACEA      CALL GOTO
44 EAB3 18A9        JR ST1X
45 EAB5
46 EAB5             ;
47 EAB5             ;      Quick disk CMD.
48 EAB5 C013EB      QUICK: CALL SIOCK      ; Sio hard check
49 EAB8 20A4        JR NZ,ST1X
50 EABA CDECEE      CALL IOFRS      ; IOCS frag reset
51 EABD 210000      LD HL,0000H      ; Frag reset
52 EAC0 223A11      LD (0DCPA),HL
53 EAC3 1A          LD A,(DE)
54 EAC4 FE4C        CP 'L'      ; Load QD
55 EAC6 CAA7EE      JP Z,QL
56 EAC9 FE53        CP 'S'      ; Save QD
57 EACB CA2EEF      JP Z,QS
58 EACE FE43        CP 'C'      ; Copy QD
59 EAD0 CA2CF1      JP Z,QC
60 EAD3 FE46        CP 'F'      ; Format QD

```

```

01 EAD5 CAB5F0      JP Z,QF
02 EAD8 FE58        CP 'X'      ; Xfer (CMTQD)
03 EADA CAA2F1      JP Z,QX
04 EADD FE44        CP 'D'      ; Directry
05 EADF CAEEF       JP Z,QD
06 EAE2 C35EEA      JST1X: JP ST1X
07 EAE5 0EF8        RAMBRD: LD C,F8H
08 EAE7 CDBAE7      CALL R12CK1
09 EAEA 3007        JR NC,RMBD1
10 EAEC 0EA8        LD C,A8H
11 EAEF CDBAE7      CALL R12CK1
12 EAF1 38EF        JR C,JST1X
13 EAF3 1A          RMBD1: LD A,(DE)
14 EAF4 FE42        CP 'B'
15 EAF6 CACAE6      JP Z,1R12RD      ; 1R12 boot-up
16 EAF9 FE53        CP 'S'
17 EAFB CA45E7      JP Z,1R12WR      ; 1R12 Write
18 EAFE 18E2        JR JST1X
19 EB00 1A          FDCK: LD A,(DE)
20 EB01 FE0D        CP 0DH
21 EB03 20DD        JR NZ,JST1X
22 EB05 C0D5E8      CALL FDCK8      ; FD i/o check
23 EB08 20D8        JR NZ,JST1X
24 EB0A C04AE4      CALL FDX
25 EB0D C09000      CALL NL
26 EB10 DF          RST 3
27 EB11 18CF        JR JST1X
28 EB13
29 EB13             ;
30 EB13             ;      Sio hard check
31 EB13 3E02        SIOCK: LD A,02H
32 EB15 D3F7        OUT (SIOBC),A
33 EB17 3EA5        LD A,A5H
34 EB19 D3F7        OUT (SIOBC),A
35 EB1B 3E02        LD A,02H
36 EB1D D3F7        OUT (SIOBC),A
37 EB1F DBF7        IN A,(SIOBC)
38 EB21 FE45        CP A5H
39 EB23 C9          RET
40 EB24
41 EB24             ;
42 EB24             ;      Error (loading)
43 EB24
44 EB24 1129F3      ?ERX: ENT
45 EB27 FE02        LD DE,MGBRK      ; 'Break!'
46 EB29 2803        CP 02H
47 EB2B 114701      JR Z,?ERX0
48 EB2E C09000      LD DE,MSGE1      ; 'Check sum err.'
49 EB31 DF          ?ERX0: CALL NL
50 EB32 18AE        RST 3
51 EB34             JR JST1X
52 EB34
53 EB34             ;
54 EB34             ;      Getline & break in check
55 EB34             ;      exit break in then jump (ST1X)
56 EB34             ;      acc=top of line data
57 EB34 E3          BGETLX: ENT
58 EB35 C1          EX (SP),HL
59 EB36 11A311      POP BC
60 EB39 C00300      LD DE,BUFER
                   CALL GETL

```

```

01 EB3C 1A          LD      A,(DE)
02 EB3D FE1B        CP      1BH
03 EB3F 28A1        JR      Z,JST1X
04 EB41 E9          JP      (HL)
05 EB42            ;
06 EB42            ; Ascii to hex convert
07 EB42            ; input (DE)=Ascii
08 EB42            ; CY=1 then jump (ST1X)
09 EB42            ;
10 EB42            HEXIYX: ENT
11 EB42 FDE3        EX      (SP),IY
12 EB44 F1          POP     AF
13 EB45 CD1004       CALL    HLHEX
14 EB48 3898        JR      C,JST1X
15 EB4A FDE9        JP      (IY)
16 EB4C            ;
17 EB4C            ; load command
18 EB4C            ;
19 EB4C            LOADX: ENT
20 EB4C CD54EB       CALL    LD00      ; Read inf & data
21 EB4F 38D3        JR      C,?ERX
22 EB51 C39DE9       JP      EXCMT     ; Exec load file
23 EB54            ;
24 EB54            ;
25 EB54            LD00: ENT
26 EB54            ;
27 EB54            ; Read information block
28 EB54            ;
29 EB54            CALL    ?RDI
30 EB57 D8          RET      C
31 EB58            ;
32 EB58            ; Disp 'Loading...'
33 EB58            ;
34 EB58            LOA0X: CALL    NL
35 EB5B 1191EE       LD      DE,MSGLD ; 'loading '
36 EB5E DF          RST      3
37 EB5F 11F110       LD      DE,NAME
38 EB62 DF          RST      3
39 EB63            ;
40 EB63            ; Load adrs set
41 EB63            ;
42 EB63 2A0411       LD      HL,(DTADR)
43 EB66 09          EXX
44 EB67 210012       LD      HL,1200H
45 EB6A 220411       LD      (DTADR),HL
46 EB6D            ;
47 EB6D            ; Read data block
48 EB6D            ;
49 EB6D C3F804       RDTBL: JP      ?R0D
50 EB70            ;
51 EB70            ;
52 EB70            ; Loading from CMT ($1200~)
53 EB70            ;
54 EB70            LDCMT: ENT
55 EB70 CD54EB       CALL    LD00
56 EB73 38AF        JR      C,?ERX
57 EB75 09          EXX
58 EB76 220411       LD      (DTADR),HL
59 EB79 187D        JR      SAC01
60 EB7B            ;

```

```

01 EB7B            ;
02 EB7B            ; Memory correction
03 EB7B            ;
04 EB7B            ;
05 EB7B            MCORX: ENT
06 EB7B CD42EB       CALL    HEXIYX
07 EB7E            MCR1X: ENT
08 EB7E C0FA05       CALL    NLPHL
09 EB81 C0B103       CALL    SPHEX
10 EB84 CD2009       CALL    ?PRTS
11 EB87 C034EB       CALL    BGETLX
12 EB8A C01004       CALL    HLHEX
13 EB8D 381B        JR      C,MCR3X
14 EB8F CDA602       CALL    .4DE
15 EB92 13          INC     DE
16 EB93 CD1F04       CALL    2HEX
17 EB96 38E6        JR      C,MCR1X
18 EB98 BE          CP      (HL)
19 EB99 20E3        JR      NZ,MCR1X
20 EB9B 13          INC     DE
21 EB9C 1A          LD      A,(DE)
22 EB9D FE0D        CP      0DH
23 EB9F 2806        JR      Z,MCR2X
24 EBA1 CD1F04       CALL    2HEX
25 EBA4 3808        JR      C,MCR1X
26 EBA6 77          LD      (HL),A
27 EBA7            MCR2X: ENT
28 EBA7 23          INC     HL
29 EBA8 18D4        JR      MCR1X
30 EBAA 60          LD      H,B
31 EBAB 69          LD      L,C
32 EBAC 18D0        JR      MCR1X
33 EBAE            ;
34 EBAE            ; Save command
35 EBAE            ;
36 EBAE CD8DEC       SAVEX: CALL    FNINP
37 EBB1 21A311       LD      HL,BUFER
38 EBB4 11F110       LD      DE,NAME
39 EBB7 011100       LD      BC,0011H
40 EBBA EDB0        LDIR
41 EBBC CDD3EC       CALL    TAINP
42 EBBF 220411       LD      (DTADR),HL ; Data adrs
43 EBC2 CDD8EC       CALL    EAINP
44 EBC5 ED4B0411     LD      BC,(DTADR)
45 EBC9 37          SCF
46 EBCA ED42        SBC     HL,BC
47 EBCC 23          INC     HL
48 EBCD 23          INC     HL
49 EBCE 220211       LD      (SIZE),HL ; Byte size
50 EB01 CDDDEC       CALL    XAINP
51 EBD4 220611       LD      (EXADR),HL ; Exec adrs
52 EBD7 3E01        LD      A,01H
53 EBD9 32F010       LD      (ATRB),A
54 EBD0 C03604       CALL    ?WRI
55 EBD0 DA24EB       JP      C,?ERX
56 EBE2 C0F3EB       CALL    SAC00
57 EBE5 C35EEA       SAVJP: JP      ST1X
58 EBE8            ;
59 EBE8            ; Save CMT ($1200~)
60 EBE8            ;

```

123

```

01 EBE8      SACMT: ENT
02 EBE8 CD3604 CALL ?WRI      ; Write information
03 EBE8 38F2  JR C,SAVJE
04 EBED 210012 LD HL,1200H
05 EBF0 220411 LD (DTADR),HL
06 EBF3 CD7504 SAC00: CALL ?WRD      ; Write data
07 EBF6 38E7  JR C,SAVJE
08 EBF8 CD0900 SAC01: CALL NL
09 EBF8 114209 LD DE,MSGOK      ; 'OK'
10 EBFE DF    RST 3
11 EBFF C9    RET
12 EC00      ;
13 EC00      ;
14 EC00      ; Verifying command
15 EC00      ;
16 EC00      ;
17 EC00      ;
18 EC00 CD8805 VRFYX: ENT
19 EC03 DA24EB CALL ?VRFY
20 EC06 C395EF VRFYE: JP C,?ERX
21 EC09      JP QDOKM      ; 'OK' MSG.
22 EC09      ;
23 EC09 CD10EC VFCMT: ENT
24 EC0C 38F5  CALL VRF120
25 EC0E 18E8  JR C,VRFYE
26 0593 P     JR SAC01      ; 'OK' MSG.
27 EC10 F3    ?VRFY0: EQU 0593H
28 EC11 D5    VRF120: DI
29 EC12 C5    PUSH DE
30 EC13 E5    PUSH BC
31 EC14 ED4B0211 LD HL,(SIZE)
32 EC18 210012 LD HL,1200H
33 EC1B C39305 JP ?VRFY0
34 EC1E      ;
35 EC1E      ; Keyin bell command 'B'
36 EC1E      ;
37 EC1E 3A9D11 SGX: LD A,(SWRK)
38 EC21 1F    RRA
39 EC22 3F    CCF
40 EC23 17    RLA
41 EC24 329D11 LD (SWRK),A
42 EC27 18BC  SGJP: JR SAVJP      ; JP ST1X
43 EC29      ;
44 EC29      ; Memory dump command
45 EC29      ;
46 EC29      ;
47 EC29 CD42EB DUMPX: ENT
48 EC2C CDA602 CALL HEXIYX
49 EC2F E5    CALL .4DE
50 EC30 CD1004 PUSH HL
51 EC33 D1    CALL HLHEX
52 EC34 3851  POP DE
53 EC36 EB    JR C,DUM1X
54 EC37 0608  EX DE,HL
55 EC39 0E17  DUM3X: LD B,08H
56 EC3B CDA005 LD C,17H
57 EC3E CDB103 CALL NLPHL
58 EC41 23    DUM2X: CALL SPHEX
59 EC42 F5    INC HL
60 EC43 3A7111 PUSH AF
        LD A,(DSPXY)

```

```

01 EC46 81    ADD A,C
02 EC47 327111 LD (DSPXY),A
03 EC4A F1    POP AF
04 EC4B FE20  CP 20H
05 EC4D 3002  JR NC,+4
06 EC4F 3E2E  LD A,2EH
07 EC51 CDB90B CALL ?ADCN
08 EC54 CD6C09 CALL PRNT3
09 EC57 3A7111 LD A,(DSPXY)
10 EC5A 0C    INC C
11 EC5B 91    SUB C
12 EC5C 327111 LD (DSPXY),A
13 EC5F 00    DEC C
14 EC60 00    DEC C
15 EC61 00    DEC C
16 EC62 E5    PUSH HL
17 EC63 ED52  SBC HL,DE
18 EC65 E1    POP HL
19 EC66 281C  JR Z,DUM1X-3
20 EC68 3EF8  LD A,F8H
21 EC6A 3200E0 LD (KEYPA),A
22 EC6D 00    NOP
23 EC6E 3A01E0 LD A,(KEYPB)
24 EC71 FEFE  CP FEH
25 EC73 2003  JR NZ,+5
26 EC75 CDA60D CALL ?BLNK
27 EC78 10C4  DJNZ DUM2X
28 EC7A CDCA08 CALL ?KEY
29 EC7D B7    OR A
30 EC7E 28FA  JR Z,-4
31 EC80 CD320A CALL ?BRK
32 EC83 20B2  JR NZ,DUM3X
33 EC85 18A0  JR SGJP      ; JP ST1X
34 EC87 21A000 DUM1X: LD HL,00A0H
35 EC8A 19    ADD HL,DE
36 EC8B 18A9  JR DUM3X-1
37 EC8D      ;
38 EC8D      ; Input file name
39 EC8D      ;
40 EC8D      ;
41 EC8D CD0900 FNINP: ENT
42 EC90 1165EE CALL NL
43 EC93 0F    LD DE,MSGSV0
44 EC94 11A311 RST 3
45 EC97 CD0300 LD DE,BUFER
46 EC9A 1A    CALL GETL
47 EC9B FE1B  LD A,(DE)
48 EC9D 2005  CP 1BH
49 EC9F 215EEA JR NZ,FNINP0
50 ECA2 E3    FNINPR: LD HL,ST1X
51 ECA3 C9    EX (SP),HL
52 ECA4      RET
53 ECA4      ;
54 ECA4      ;
55 ECA6 11AD11 FNINP0: LD B,00H
56 ECA9 21A311 LD DE,BUFER+10
57 ECAC 1A    LD HL,BUFER
58 ECAD FE0D  LD A,(DE)
59 ECAF 2820  CP 0DH
60 ECB1      JR Z,FNINP6

```

```

01 ECB1      ; Space check
02 ECB1      ;
03 ECB1 FE20  FNINP1: CP      ' '      ; Space ?
04 ECB3 2004      JR      NZ,FNINP2
05 ECB5 13      INC      DE
06 ECB6 1A      LD      A,(DE)
07 ECB7 18F8      JR      FNINP1
08 ECB9      ;
09 ECB9      ; " ?
10 ECB9      ;
11 ECB9 FE22  FNINP2: CP      22H
12 ECB8 2808      JR      Z,FNINP4
13 ECB0      ;
14 ECB0      ; Trans (DE)(HL)
15 ECB0      ;
16 ECB0 77  FNINP3: LD      (HL),A
17 ECBE 23      INC      HL
18 ECBF 04      INC      B
19 ECC0 3E11      LD      A,11H
20 ECC2 B8      CP      B
21 ECC3 28C8      JR      Z,FNINP
22 ECC5      ;
23 ECC5      ; Filename end check
24 ECC5      ;
25 ECC5 13  FNINP4: INC      DE
26 ECC6 1A      LD      A,(DE)
27 ECC7 FE22      CP      22H      ; " ?
28 ECC9 2804      JR      Z,FNINP5
29 ECCB FE00      CP      0DH
30 ECCD 20EE      JR      NZ,FNINP3
31 ECCF 3E00  FNINP5: LD      A,0DH
32 ECD1 77  FNINP6: LD      (HL),A
33 ECD2 C9      RET
34 ECD3      ;
35 ECD3      ; Input save condition
36 ECD3      ;
37 ECD3  TAINP: ENT      DE,MSGTA      ; 'Top adrs?'
38 ECD3 1170EE      LD      AINP0
39 ECD6 1808      JR      AINP0
40 ECD8  EAINP: ENT      DE,MSGEA      ; 'End adrs?'
41 ECD8 117BEE      LD      AINP0
42 ECD8 1803      JR      AINP0
43 ECDD  XAINP: ENT      DE,MSGXA      ; 'Exc adrs?'
44 ECDD 1186EE      LD      AINP0
45 ECE0 CD0900      CALL      NL
46 ECE3 DF      RST      3
47 ECE4 D5      PUSH     DE
48 ECE5 11A311      LD      DE,BUFER
49 ECE8 CD0300      CALL      GETL
50 ECEB 1A      LD      A,(DE)
51 ECEC FE1B      CP      1BH      ; Break ?
52 ECEE D1      POP      DE
53 ECEF 28AE      JR      Z,FNINPR
54 ECF1 D5      PUSH     DE
55 ECF2 11AD11      LD      DE,BUFER+10
56 ECF5 CD1004      CALL      HLHEX
57 ECF8 D1      POP      DE
58 ECF9 38E5      JR      C,AINP0
59 ECFB C9      RET
60 ECFC      ;

```

```

01 ECFC      ; Exec file
02 ECFC      ; IN HL ; Data size point
03 ECFC      ; BC ; Default code
04 ECFC      ;
05 ECFC      ; EXF: ENT
06 ECFC DBCE      IN      A,(CEH)      ; Mode sw.
07 ECFC CB4F      BIT      1,A
08 ED00 2806      JR      Z,70MOD
09 EB02 AF      XOR      A
10 ED03 D3CE      OUT      (CEH),A      ; 800 Mode set
11 ED05 CDE1E8      CALL     PLTST      ; Pallet set
12 ED08      ;
13 ED08 4E      70MOD: LD      C,(HL)      ; BCDData adrs
14 ED09 23      INC      HL
15 ED0A 46      LD      B,(HL)
16 ED0B 23      INC      HL
17 ED0C 5E      LD      E,(HL)      ; DEData adrs
18 ED0D 7E      LD      A,(HL)
19 ED0E 23      INC      HL
20 ED0F 56      LD      D,(HL)
21 ED10 B6      OR      (HL)
22 ED11 D5      PUSH     DE      ; HLExec adrs
23 ED12 23      INC      HL
24 ED13 5E      LD      E,(HL)
25 ED14 B6      OR      (HL)
26 ED15 23      INC      HL
27 ED16 56      LD      D,(HL)
28 ED17 B6      OR      (HL)
29 ED18 D5      PUSH     DE
30 ED19 DDE1      POP      IX
31 ED1B D1      POP      DE
32 ED1C 2002      JR      NZ,BLKTR      ; Data adrs & Exec adrs
33 ED1E D3E0      OUT      (E0H),A      ; Bank $0000~8000 Dram
34 ED20      ;
35 ED20 210012  BLKTR: LD      HL,1200H
36 ED23 3E12      LD      A,12H
37 ED25 BA      CP      D
38 ED26 3806      JR      C,LDDEC
39 ED28 2014      JR      NZ,LDINC
40 ED2A AF      XOR      A
41 ED2B BB      CP      E
42 ED2C 3010      JR      NC,LDINC
43 ED2E 0B      LDDEC: DEC      BC
44 ED2F E5      PUSH     HL
45 ED30 210000      LD      HL,0000H
46 ED33 19      ADD      HL,DE
47 ED34 09      ADD      HL,BC
48 ED35 E5      PUSH     HL
49 ED36 D1      POP      DE
50 ED37 E1      POP      HL
51 ED38 09      ADD      HL,BC
52 ED39 03      INC      BC
53 ED3A EDB8      LDDR      BC
54 ED3C 1802      JR      EXF0
55 ED3E      ;
56 ED3E EDB0      LDINC: LDIR      ; BCD default code
57 ED40 D9      EXF0: EXX      (IX)
58 ED41 DDE9      JP
59 ED43      ;
60 ED43      ;

```

```

01 ED43          ;
02 ED43 50      SELM0: DEFB 50H          ; 'Please push key'
03 ED44 B8      DEFB B8H              ; L
04 ED45 92      DEFB 92H              ; E
05 ED46 A1      DEFB A1H              ; A
06 ED47 A4      DEFB A4H              ; S
07 ED48 92      DEFB 92H              ; E
08 ED49 20      DEFB 20H              ;
09 ED4A 9E      DEFB 9EH              ; P
10 ED4B A5      DEFB A5H              ; U
11 ED4C A4      DEFB A4H              ; S
12 ED4D 98      DEFB 98H              ; H
13 ED4E 20      DEFB 20H              ;
14 ED4F A9      DEFB A9H              ; K
15 ED50 92      DEFB 92H              ; E
16 ED51 8D      DEFB 8DH              ; Y
17 ED52 0D      DEFB 0DH              ;
18 ED53          ;
19 ED53 463A46  SELM1: DEFM 'F:F'        ; 'F:Floppy disk'
20 ED56 B8      DEFB B8H              ; L
21 ED57 B7      DEFB B7H              ; O
22 ED58 9E      DEFB 9EH              ; P
23 ED59 9E      DEFB 9EH              ; P
24 ED5A 8D      DEFB 8DH              ; Y
25 ED5B 20      DEFB 20H              ;
26 ED5C 9C      DEFB 9CH              ; D
27 ED5D A6      DEFB A6H              ; I
28 ED5E A4      DEFB A4H              ; S
29 ED5F A9      DEFB A9H              ; K
30 ED60 0D      DEFB 0DH              ;
31 ED61          ;
32 ED61 513A51  SELM2: DEFM 'Q:Q'        ; 'Q:Quick disk'
33 ED64 A5      DEFB A5H              ; U
34 ED65 A6      DEFB A6H              ; I
35 ED66 9F      DEFB 9FH              ; C
36 ED67 A9      DEFB A9H              ; K
37 ED68 20      DEFB 20H              ;
38 ED69 9C      DEFB 9CH              ; D
39 ED6A A6      DEFB A6H              ; I
40 ED6B A4      DEFB A4H              ; S
41 ED6C A9      DEFB A9H              ; K
42 ED6D 0D      DEFB 0DH              ;
43 ED6E          ;
44 ED6E 433A43  SELM3: DEFM 'C:C'        ; 'C:Cassette tape'
45 ED71 A1      DEFB A1H              ; A
46 ED72 A4      DEFB A4H              ; S
47 ED73 A4      DEFB A4H              ; S
48 ED74 92      DEFB 92H              ; E
49 ED75 96      DEFB 96H              ; T
50 ED76 96      DEFB 96H              ; T
51 ED77 92      DEFB 92H              ; E
52 ED78 20      DEFB 20H              ;
53 ED79 96      DEFB 96H              ; T
54 ED7A A1      DEFB A1H              ; A
55 ED7B 9E      DEFB 9EH              ; P
56 ED7C 92      DEFB 92H              ; E
57 ED7D 0D      DEFB 0DH              ;
58 ED7E          ;
59 ED7E 4D3A4D  SELM4: DEFM 'M:M'        ; 'M:Monitor'
60 ED81 B7      DEFB B7H              ; O

```

```

01 ED82 B0      DEFB B0H              ; N
02 ED83 A6      DEFB A6H              ; I
03 ED84 96      DEFB 96H              ; T
04 ED85 B7      DEFB B7H              ; O
05 ED86 9D      DEFB 9DH              ; R
06 ED87 0D      DEFB 0DH              ;
07 ED88          ;
08 ED88          ; IPLM0: ENT
09 ED88 49504C20 DEFM 'IPL '          ; 'IPL is loading'
10 ED8C A6      DEFB A6H              ; I
11 ED8D A4      DEFB A4H              ; S
12 ED8E 20      DEFB 20H              ;
13 ED8F B8      DEFB B8H              ; L
14 ED90 B7      DEFB B7H              ; O
15 ED91 A1      DEFB A1H              ; A
16 ED92 9C      DEFB 9CH              ; D
17 ED93 A6      DEFB A6H              ; I
18 ED94 B0      DEFB B0H              ; N
19 ED95 97      DEFB 97H              ; G
20 ED96 20      DEFB 20H              ;
21 ED97 0D      DEFB 0DH              ;
22 ED98 4D      DEFM 'M'              ; 'Make ready CMT'
23 ED99 A1      DEFB A1H              ; A
24 ED9A A9      DEFB A9H              ; K
25 ED9B 92      DEFB 92H              ; E
26 ED9C 20      DEFB 20H              ;
27 ED9D 9D      DEFB 9DH              ; R
28 ED9E 92      DEFB 92H              ; E
29 ED9F A1      DEFB A1H              ; A
30 EDA0 9C      DEFB 9CH              ; D
31 EDA1 B0      DEFB B0H              ; Y
32 EDA2 20434D54 DEFM ' CMT'          ;
33 EDA6 0D      DEFB 0DH              ;
34 EDA7 4D      DEFM 'M'              ; 'Make ready QD'
35 EDA8 A1      DEFB A1H              ; A
36 EDA9 A9      DEFB A9H              ; K
37 EDAA 92      DEFB 92H              ; E
38 EDAB 20      DEFB 20H              ;
39 EDAC 9D      DEFB 9DH              ; R
40 EDAD 92      DEFB 92H              ; E
41 EDAE A1      DEFB A1H              ; A
42 EDAF 9C      DEFB 9CH              ; D
43 EDB0 B0      DEFB B0H              ; Y
44 EDB1 205144  DEFM ' QD'            ;
45 EDB4 0D      DEFB 0DH              ;
46 EDB5          ; IPLM3: ENT
47 EDB5 4D      DEFM 'M'              ; 'Make ready FD'
48 EDB6 A1      DEFB A1H              ; A
49 EDB7 A9      DEFB A9H              ; K
50 EDB8 92      DEFB 92H              ; E
51 EDB9 20      DEFB 20H              ;
52 EDBA 9D      DEFB 9DH              ; R
53 EDBB 92      DEFB 92H              ; E
54 EDBC A1      DEFB A1H              ; A
55 EDBD 9C      DEFB 9CH              ; D
56 EDBE B0      DEFB B0H              ; Y
57 EDBF 204644  DEFM ' FD'            ;
58 EDC2 0D      DEFB 0DH              ;
59 EDC3 20202020 DEFM ' IPL '          ; 'IPL is looking for ..'
60 EDC7 2049504C

```

```

01 EDCB 20
02 EDCC A6      DEFB A6H      ; I
03 EDCD A4      DEFB A4H      ; S
04 EDCE 20      DEFB 20H      ;
05 EDCF B8      DEFB B8H      ; L
06 EDD0 B7      DEFB B7H      ; O
07 EDD1 B7      DEFB B7H      ; O
08 EDD2 A9      DEFB A9H      ; K
09 EDD3 A6      DEFB A6H      ; I
10 EDD4 B0      DEFB B0H      ; N
11 EDD5 97      DEFB 97H      ; G
12 EDD6 20      DEFB 20H      ;
13 EDD7 AA      DEFB AAH      ; F
14 EDD8 B7      DEFB B7H      ; O
15 EDD9 9D      DEFB 9DH      ; R
16 EDDA 20      DEFB 20H      ;
17 EDD8 A1      DEFB A1H      ; A
18 EDDC 20      DEFB 20H      ;
19 EDD0 9E      DEFB 9EH      ; P
20 EDD8 9D      DEFB 9DH      ; R
21 EDDF B7      DEFB B7H      ; O
22 EDE0 97      DEFB 97H      ; G
23 EDE1 9D      DEFB 9DH      ; R
24 EDE2 A1      DEFB A1H      ; A
25 EDE3 B3      DEFB B3H      ; M
26 EDE4 0D      DEFB 0DH      ;
27 EDE5      ;
ERRM0: ENT      DEFM 'FD:L'      ; 'FD:Loading error'
28 EDE5 46443A4C DEFB B7H      ; O
29 EDE9 B7      DEFB B7H      ; O
30 EDEA A1      DEFB A1H      ; A
31 EDEB 9C      DEFB 9CH      ; D
32 EDEC A6      DEFB A6H      ; I
33 EDED B0      DEFB B0H      ; N
34 EDEE 97      DEFB 97H      ; G
35 EDEF 20      DEFB 20H      ;
36 EDF0 92      DEFB 92H      ; E
37 EDF1 9D      DEFB 9DH      ; R
38 EDF2 9D      DEFB 9DH      ; R
39 EDF3 B7      DEFB B7H      ; O
40 EDF4 9D      DEFB 9DH      ; R
41 EDF5 0D      DEFB 0DH      ;
42 EDF6      ;
ERRM1: ENT      DEFM 'FD:N'      ; 'FD:Not master'
44 EDF6 46443A4E DEFB B7H      ; O
45 EDFA B7      DEFB B7H      ; O
46 EDFB 96      DEFB 96H      ; T
47 EDFC 20      DEFB 20H      ;
48 EDFD B3      DEFB B3H      ; M
49 EDFE A1      DEFB A1H      ; A
50 EDFF A4      DEFB A4H      ; S
51 EE00 96      DEFB 96H      ; T
52 EE01 92      DEFB 92H      ; E
53 EE02 9D      DEFB 9DH      ; R
54 EE03 0D      DEFB 0DH      ;
55 EE04      ;
ERRM2: DEFM      'CMT:L'      ; 'CMT:Loading error'
56 EE04 434D543A DEFB B7H      ; O
57 EE08 4C      DEFB A1H      ; A
58 EE09 B7      DEFB 9CH      ; D
59 EE0A A1      DEFB 9CH      ; D
60 EE0B 9C

```

```

01 EE0C A6      DEFB A6H      ; I
02 EE0D B0      DEFB B0H      ; N
03 EE0E 97      DEFB 97H      ; G
04 EE0F 20      DEFB 20H      ;
05 EE10 92      DEFB 92H      ; E
06 EE11 9D      DEFB 9DH      ; R
07 EE12 9D      DEFB 9DH      ; R
08 EE13 B7      DEFB B7H      ; O
09 EE14 9D      DEFB 9DH      ; R
10 EE15 0D      DEFB 0DH      ;
11 EE16      ;
ERRM3: DEFM      'QD:L'      ; 'QD:Loading error'
12 EE16 51443A4C DEFB B7H      ; O
13 EE1A B7      DEFB A1H      ; A
14 EE1B A1      DEFB 9CH      ; D
15 EE1C 9C      DEFB A6H      ; I
16 EE1D A6      DEFB B0H      ; N
17 EE1E B0      DEFB 97H      ; G
18 EE1F 97      DEFB 20H      ;
19 EE20 20      DEFB 92H      ; E
20 EE21 92      DEFB 9DH      ; R
21 EE22 9D      DEFB 9DH      ; R
22 EE23 9D      DEFB 9DH      ; R
23 EE24 B7      DEFB B7H      ; O
24 EE25 9D      DEFB 9DH      ; R
25 EE26 0D      DEFB 0DH      ;
26 EE27      ;
ERRM4: DEFM      'QD:F'      ; 'QD:File mode error'
27 EE27 51443A46 DEFB A6H      ; I
28 EE2B A6      DEFB B8H      ; L
29 EE2C B8      DEFB 92H      ; E
30 EE2D 92      DEFB 20H      ;
31 EE2E 20      DEFB B3H      ; M
32 EE2F B3      DEFB B7H      ; O
33 EE30 B7      DEFB 9CH      ; D
34 EE31 9C      DEFB 92H      ; E
35 EE32 92      DEFB 20H      ;
36 EE33 20      DEFB 92H      ; E
37 EE34 92      DEFB 9DH      ; R
38 EE35 9D      DEFB 9DH      ; R
39 EE36 9D      DEFB 9DH      ; R
40 EE37 B7      DEFB B7H      ; O
41 EE38 9D      DEFB 9DH      ; R
42 EE39 0D      DEFB 0DH      ;
43 EE3A      ;
ERRM5: DEFM      'SRAM:C'      ; 'SRAM:Check sum error'
44 EE3A 5352414D DEFB 98H      ; H
45 EE3E 3A43      DEFB 92H      ; E
46 EE40 98      DEFB 9FH      ; C
47 EE41 92      DEFB A9H      ; K
48 EE42 9F      DEFB 20H      ;
49 EE43 A9      DEFB A4H      ; S
50 EE44 20      DEFB A5H      ; U
51 EE45 A4      DEFB B3H      ; M
52 EE46 A5      DEFB 20H      ;
53 EE47 B3      DEFB 92H      ; E
54 EE48 20      DEFB 9DH      ; R
55 EE49 92      DEFB 9DH      ; R
56 EE4A 9D      DEFB 9DH      ; R
57 EE4B 9D      DEFB 0DH      ;
58 EE4C 0D      DEFB 0DH      ;
59 EE4D      ;
MONMSG: DEFM      '... MONITOR 92-504M ...'
60 EE4D 2A2A2020

```

```

01 EE51 4D4F4E49
02 EE55 544F5220
03 EE59 395A2D35
04 EE5D 30344D20
05 EE61 202A2A
06 EE64 0D
07 EE65 46
08 EE66 A6
09 EE67 B8
10 EE68 92
11 EE69 B0
12 EE6A A1
13 EE6B B3
14 EE6C 92
15 EE6D 3F
16 EE6E 20
17 EE6F 0D
18 EE70 54
19 EE71 B7
20 EE72 9E
21 EE73 20
22 EE74 A1
23 EE75 9C
24 EE76 9D
25 EE77 A4
26 EE78 3F
27 EE79 20
28 EE7A 0D
29 EE7B 45
30 EE7C B0
31 EE7D 9C
32 EE7E 20
33 EE7F A1
34 EE80 9C
35 EE81 9D
36 EE82 A4
37 EE83 3F
38 EE84 20
39 EE85 0D
40 EE86 45
41 EE87 9B
42 EE88 9F
43 EE89 20
44 EE8A A1
45 EE8B 9C
46 EE8C 9D
47 EE8D A4
48 EE8E 3F
49 EE8F 20
50 EE90 0D
51 EE91
52 EE91 4C
53 EE92 B7
54 EE93 A1
55 EE94 9C
56 EE95 A6
57 EE96 B0
58 EE97 97
59 EE98 20
60 EE99 0D
MSGSV0: DEFB 0DH
         DEFB 'F'
         DEFB A6H
         DEFB B8H
         DEFB 92H
         DEFB B0H
         DEFB A1H
         DEFB B3H
         DEFB 92H
         DEFB 3FH
         DEFB 20H
         DEFB 0DH
MSGTA:  DEFB 'T'
         DEFB B7H
         DEFB 9EH
         DEFB 20H
         DEFB A1H
         DEFB 9CH
         DEFB 9DH
         DEFB A4H
         DEFB 3FH
         DEFB 20H
         DEFB 0DH
MSGEA:  DEFB 'E'
         DEFB B0H
         DEFB 9CH
         DEFB 20H
         DEFB A1H
         DEFB 9CH
         DEFB 9DH
         DEFB A4H
         DEFB 3FH
         DEFB 20H
         DEFB 0DH
MSGXA:  DEFB 'E'
         DEFB 9BH
         DEFB 9FH
         DEFB 20H
         DEFB A1H
         DEFB 9CH
         DEFB 9DH
         DEFB A4H
         DEFB 3FH
         DEFB 20H
         DEFB 0DH
MSGLD:  ENT
         DEFB 'L'
         DEFB B7H
         DEFB A1H
         DEFB 9CH
         DEFB A6H
         DEFB B0H
         DEFB 97H
         DEFB 20H
         DEFB 0DH
; 'Filename? '
; I
; L
; E
; N
; A
; M
; E
; ?
;
; 'Top adrs? '
; O
; P
;
; A
; D
; R
; S
; ?
;
; 'End adrs? '
; N
; D
;
; A
; D
; R
; S
; ?
;
; 'Exc adrs? '
; X
; C
;
; A
; D
; R
; S
; ?
;
; 'Loading '
; O
; A
; D
; I
; N
; G

```

```

01 EE9A
02 EE9A
03 EE9A
04 EE9A 00
05 EE9B CF
06 EE9C 3F
07 EE9D 07
08 EE9E
09 EE9E 00
10 EE9F CF
11 EEA0 00
12 EEA1 07
13 EEA2
14 EEA2
15 EEA2
16 EEA2
17 EEA2 00
18 EEA3 10
19 EEA4 20
20 EEA5 30
21 EEA6 40
22 EEA7
23 EEA7
;
; Pio initialize data
;
PIOIDA: DEFB 00H
         DEFB CFH
         DEFB 3FH
         DEFB 07H
; Int vector
; Mode 3 (bit)
; I/O Reg. set
; Int sequence
;
PIOIDB: DEFB 00H
         DEFB CFH
         DEFB 00H
         DEFB 07H
; Int vector
; Mode 3 (bit)
; I/O Reg. set
; Int sequence
;
; Pallet set data
;
PLTDT:  DEFB 00H
         DEFB 10H
         DEFB 20H
         DEFB 30H
         DEFB 40H
; all black
; Sw 0,1=0
; SKP H

```

```

01 EEA7 ;
02 EEA7 ;
03 EEA7 ; Quick disk control command
04 EEA7 ;
05 EEA7 ;   QL=Quick load
06 EEA7 ;   QS=Quick save
07 EEA7 ;   QD=Quick directory
08 EEA7 ;   QF=Quick formatting
09 EEA7 ;   QX=Quick xfer (CMTDisk)
10 EEA7 ;   QC=Quick copy (DiskDisk)
11 EEA7 ;
12 EEA7 ;
13 11A3 P BUFF: EQU 11A3H ; Keyin buffer
14 EEA7 ;
15 EEA7 ; =====
16 EEA7 ; QUICK DISK LOAD COMMAND
17 EEA7 ;
18 EEA7 ; =====
19 EEA7 ;
20 EEA7 ;
21 EEA7 ;
22 EEA7 CD27EF QL: ENT
23 EEA7 3834 CALL QDRCK ; Ready check
24 EEA7 CD8DEC JR C, QER00
25 EEA7 CD5FF2 CALL FNINP ; Input filename
26 EEB2 CALL HDPCL ; Head point clear
27 EEB2 ;
28 EEB2 ; Disp 'Loading...'
29 EEB2 1191EE LD DE, MSGLD ; 'Loading'
30 EEB5 DF RST 3
31 EEB6 ;
32 EEB6 ; File search
33 EEB6 ;
34 EEB6 CDF7EE FILESH: CALL FILSCH
35 EEB9 3825 JR C, QER00
36 EEBB ;
37 EEBB ; Attribute check
38 EEBB ;
39 EEBB 3AF010 QDATRC: LD A, (ATRB)
40 EEBE FE01 CP OBJCD
41 EEC0 20F4 JR NZ, FILESH
42 EEC2 ;
43 EEC2 ; Disp 'Loading ...'
44 EEC2 ;
45 EEC2 ;
46 EEC2 11F110 DSFLNA: ENT
47 EEC5 DF LD DE, NAME
48 EEC6 RST 3
49 EEC6 ;
50 EEC6 ; Iocs parameter set
51 EEC6 210012 LD HL, 1200H
52 EEC9 1803 JR LPARA1
53 EECB 2A0611 LPARA0: LD HL, (QDTADR)
54 EEC6 223211 LPARA1: LD (QDPC), HL ; Data adrs set
55 EED1 2A0411 LD HL, (QSIZE)
56 EED4 223411 LD (QDPE), HL ; Read data block cmd.
57 EED7 210301 LD HL, 0103H
58 EEDA 223011 LD (QDPA), HL
59 EEDD ;
60 EEDD ; Read data block

```

```

01 EEDD ;
02 EEDD CD10E0 ; CALL QDIOS ; QD iocs
03 EEE0 384F QER00: JR C, QER01
04 EEE2 ;
05 EEE2 ; Exec load file
06 EEE2 ;
07 EEE2 010003 LD BC, 0300H ; Default code (QD)
08 EEE5 D9 EXX
09 EEE6 210411 LD HL, QSIZE
10 EEE9 C3FCEC JP EXF
11 EEEC ;
12 EEEC ; Iocs flag reset
13 EEEC ;
14 EEEC ;
15 EEEC AF IOFRS: ENT
16 EEE0 324411 XOR A ; Acc00H
17 EEF0 323F11 LD (MTF), A ; Motor flag
18 EEF3 324111 LD (FNUPS), A ; File number flag
19 EEF6 C9 LD (FNUPF), A ; File number up flag
20 EEF7 RET
21 EEF7 ;
22 EEF7 ; File search sub.
23 EEF7 ;
24 EEF7 ;
25 EEF7 FILSCH: ENT
26 EEF7 ;
27 EEF7 ; Iocs parameter set
28 EEF7 ;
29 EEF7 210300 LD HL, 0003H ; Read information cmd.
30 EEEA 223011 LD (QDPA), HL
31 EEF0 21F010 LD HL, 10F0H ; Head adrs
32 EEF0 223211 LD (QDPC), HL
33 EEF0 214000 LD HL, 0040H ; Read size
34 EEF0 223411 LD (QDPE), HL
35 EEF0 ;
36 EEF0 ; Read information block
37 EEF0 ;
38 EEF0 CD10E0 QLINF: CALL QDIOS
39 EEF0 D8 RET C
40 EEF0 ;
41 EEF0 ; File name check
42 EEF0 ;
43 EEF0 3AA311 LD A, (BUFF)
44 EEF0 FE0D CP 0DH
45 EEF2 C8 RET Z ; CY=0
46 EEF3 21A311 LD HL, BUFF
47 EEF6 11F110 LD DE, NAME
48 EEF9 0611 LD B, NAMSIZ
49 EEFB 1A LD A, (DE)
50 EEF0 BE CP (HL)
51 EEF1 20EA JR NZ, QLINF
52 EEF1 FE0D CP 0DH
53 EEF2 C8 RET Z ; CY=0
54 EEF2 13 INC DE
55 EEF2 23 INC HL
56 EEF2 10F5 DJNZ LDFNCK
57 EEF2 C9 RET
58 EEF2 ;
59 EEF2 ; Quick disk ready check
60 EEF2 ;

```

```

01 EF27      QDRCK: ENT
02 EF27 AF    XOR    A
03 EF28 323111 LD     (QDPB),A
04 EF2B C3E6EF JP      QDRWC0
05 EF2E      ;
06 EF2E      ;=====
07 EF2E      ;
08 EF2E      ; Quick disk save command
09 EF2E      ;
10 EF2E      ;=====
11 EF2E      ;
12 EF2E      QS: ENT
13 EF2E CDE1EF CALL   QDRWCK      ; Ready & wrpt check
14 EF31 385F   JR      C,QER03
15 EF33      ;
16 EF33      ; Input file name
17 EF33      ;
18 EF33 CD8DEC CALL   FNINP      ; Input file name
19 EF36 3AA311 LD     A,(BUFF)
20 EF39 FE0D   CP     0DH
21 EF3B 28F1   JR      Z,QS
22 EF3D 21A311 LD     HL,BUFF
23 EF40 11F110 LD     DE,NAME
24 EF43 011100 LD     BC,0011H
25 EF46 EDB0   LDIR
26 EF48      ;
27 EF48      ; Input top & end & exc adrs
28 EF48      ;
29 EF48 CDD3EC CALL   TAINP      ; Top adrs
30 EF4B 220611 LD     (QDTADR),HL
31 EF4E CDD8EC CALL   EAINP      ; End adrs
32 EF51 B7     OR     A      ; Carry reset
33 EF52 ED4B0611 LD     BC,(QDTADR)
34 EF56 ED42   SBC     HL,BC
35 EF58 23     INC     HL
36 EF59 220411 LD     (QSIZE),HL
37 EF5C CDDDEC CALL   XAINP      ; Exc adrs
38 EF5F 220811 LD     (QEXADR),HL
39 EF62      ;
40 EF62      ; Attribute set
41 EF62      ;
42 EF62 3E01   LD     A,OBJCD      ; OBJ Code
43 EF64 32F010 LD     (ATRB),A
44 EF67      ;
45 EF67      ; File end search
46 EF67      ;
47 EF67 CD9FEF CALL   FEDSCH
48 EF6A 3826   JR      C,QER03
49 EF6C      ;
50 EF6C      ; Save file start
51 EF6C      ;
52 EF6C FE28   CP     NTFECD      ; Not found ?
53 EF6E C202F2 JP      NZ,QERTRT
54 EF71      ;
55 EF71      ; Iocs parameter set
56 EF71      ;
57 EF71 2A0611 QSIOST: LD     HL,(QDTADR)      ; Data top adrs
58 EF74 223611 QSVIOS: LD     (QDPG),HL      ; Copy jump point
59 EF77 210404 LD     HL,0404H      ; Write Inf+Data blk cmd
60 EF7A 223011 LD     (QDPA),HL

```

```

01 EF7D 21F010 LD     HL,ATRB      ; Infm top adrs
02 EF80 223211 LD     (QDPC),HL
03 EF83 214000 LD     HL,0040H      ; Infm byte size
04 EF86 223411 LD     (QDPE),HL
05 EF89 2A0411 LD     HL,(QSIZE)      ; Data byte size
06 EF8C 223811 LD     (QDPI),HL
07 EF8F      ;
08 EF8F      ; Write inf & data block
09 EF8F      ;
10 EF8F CD10E0 CALL   QDIOS
11 EF92 DA28F0 QER03: JP      C,QER04
12 EF95      ;
13 EF95      ; Quick disk 'OK' message
14 EF95      ;
15 EF95      QOOKM: ENT
16 EF95 CD0900 CALL   NL
17 EF98 1168F2 LD     DE,MSGQOK      ; 'OK!'
18 EF9B DF     RST     3
19 EF9C C35EEA JP      ST1X
20 EF9F      ;
21 EF9F      ; File end search sub.
22 EF9F      ;
23 EF9F      FEDSCH: ENT
24 EF9F CD5FF2 CALL   HDPCL      ; Hard point clear
25 EFA2      ;
26 EFA2      ; Iocs parameter set
27 EFA2      ;
28 EFA2 210300 LD     HL,0003H      ; Read information cmd.
29 EFA5 223011 LD     (QDPA),HL
30 EFA8 21A311 LD     HL,BUFF
31 EFAB 223211 LD     (QDPC),HL
32 EFAE 214000 LD     HL,0040H      ; Read size
33 EF81 223411 LD     (QDPE),HL
34 EFB4      ;
35 EFB4      ; Too many files ?
36 EFB4      ;
37 EFB4 AF     XOR     A
38 EFB5 323C11 LD     (QDCPC),A
39 EFB8 3A3C11 LD     A,(QDCPC)
40 EFB8 3C     INC     A
41 EFB8 FE21   CP     33      ; Files
42 EFB8 323C11 LD     (QDCPC),A
43 EFC1 3E33   LD     A,TMFECD      ; Too many files err
44 EFC3 D0     RET     NC
45 EFC4      ;
46 EFC4      ; End file check
47 EFC4      ;
48 EFC4 CD10E0 CALL   QDIOS
49 EFC7 3F     CCF
50 EFC8 D0     RET     NC
51 EFC9      ;
52 EFC9      ; Same file name check
53 EFC9      ;
54 EFC9 11A411 LD     DE,BUFF+1
55 EFC9 21F110 LD     HL,NAME
56 EFCF 0611   LD     B,11H
57 EFD1 1A     LD     A,(DE)
58 EFD2 BE     CP     (HL)
59 EFD3 20E3   JR      NZ,EDFLCK
60 EFD5 FE0D   CP     0DH

```

```

01 EFD7 2804      JR      Z,SFNC0
02 EFD9 13        INC     DE
03 EFDA 23        INC     HL
04 EFD8 10F4      DJNZ    SFNC
05 EFDD 3E2A      SFNC0: LD      A,ALEXCD      ; Already exist
06 EFD7 37        SCF          ; cy=1
07 EFEE C9        RET
08 EFEE C9
09 EFEE C9
10 EFEE C9
11 EFEE C9
12 EFEE C9
13 EFEE C9
14 EFEE C9
15 EFEE C9
16 EFEE C9
17 EFEE C9
18 EFEE C9
19 EFEE C9
20 EFEE C9
21 EFEE C9
22 EFEE C9
23 EFEE C9
24 EFEE C9
25 EFEE C9
26 EFEE C9
27 EFEE C9
28 EFEE C9
29 EFEE C9
30 EFEE C9
31 EFEE C9
32 EFEE C9
33 EFEE C9
34 EFEE C9
35 EFEE C9
36 EFEE C9
37 EFEE C9
38 EFEE C9
39 EFEE C9
40 EFEE C9
41 EFEE C9
42 EFEE C9
43 EFEE C9
44 EFEE C9
45 EFEE C9
46 EFEE C9
47 EFEE C9
48 EFEE C9
49 EFEE C9
50 EFEE C9
51 EFEE C9
52 EFEE C9
53 EFEE C9
54 EFEE C9
55 EFEE C9
56 EFEE C9
57 EFEE C9
58 EFEE C9
59 EFEE C9
60 EFEE C9

```

Ready & write protect check

```

QDRWCK: LD      A,FFH
        LD      (QDPB),A
QDRWC0: LD      A,01H
        LD      (QDPA),A
        CALL    QDIOS
        RET

```

Quick disk directory command

```

QD:      ENT     QDRCK      ; Ready check
        CALL    JR      C,QER04
        CALL    HDPCL      ; Hard point clear
        LD      B,00H      ; Counter reset

```

Disp 'Directory of QD:'

```

        CALL    NL
        LD      DE,DIRMSG
        RST     3

```

locs parameter set

```

        LD      HL,QDIRBF      ; QD dir buffer
DIRIOP: LD      (QDPC),HL
        LD      HL,0003H      ; Read information cmd.
        LD      (QDPA),HL
        LD      HL,0040H      ; Read size
        LD      (QDPE),HL

```

Read information block

```

        PUSH    BC
        CALL    QDIOS
        POP     BC
        JR      C,DIREFC
        INC     B

```

Buffer adrs increment

```

        LD      HL,(QDPC)
        LD      DE,0012H
        ADD     HL,DE
        JR      DIRIOP

```

End file check

```

01 F023
02 F023 FE28      DIREFC: CP      NTFECD      ; Not found ?
03 F025 2804      JR      Z,DIRMTF
04 F027 37        SCF
05 F028 DAC9F0    QER04: JP      C,QER05
06 F02B
07 F02B
08 F02B
09 F02B 3E06      DIRMTF: LD      A,06H      ; Motor off cmd.
10 F02D 323011    LD      (QDPA),A
11 F030 C5        PUSH    BC
12 F031 CD10E0    CALL    QDIOS
13 F034 C1        POP     BC
14 F035
15 F035
16 F035
17 F035 AF        XOR     A
18 F036 B8        CP      B
19 F037 3079      JR      NC,QDOKM0
20 F039
21 F039
22 F039
23 F039 CD0900    CALL    NL
24 F03C 2190CD    LD      HL,QDIRBF      ; QD dir buffer
25 F03F
26 F03F
27 F03F
28 F03F 7E        DSPATR: LD      A,(HL)
29 F040 116CF2    LD      DE,MSGQ01      ; 'OBJ'
30 F043 3D        DEC     A
31 F044 2836      JR      Z,DRDIS0      ; Object file
32 F046 1170F2    LD      DE,MSGQ02      ; 'BTX'
33 F049 3D        DEC     A
34 F04A 2830      JR      Z,DRDIS0      ; Btx file
35 F04C 1174F2    LD      DE,MSGQ03      ; 'BSD'
36 F04F 3D        DEC     A
37 F050 282A      JR      Z,DRDIS0      ; Bsd file
38 F052 1178F2    LD      DE,MSGQ04      ; 'BRD'
39 F055 3D        DEC     A
40 F056 2824      JR      Z,DRDIS0      ; Brd file
41 F058 117CF2    LD      DE,MSGQ05      ; 'RB '
42 F05B 3D        DEC     A
43 F05C 281E      JR      Z,DRDIS0      ; Rb file
44 F05E 3D        DEC     A
45 F05F 2818      JR      Z,DRDIS1      ; '???' file
46 F061 1180F2    LD      DE,MSGQ07      ; 'LIB'
47 F064 3D        DEC     A
48 F065 2815      JR      Z,DRDIS0      ; Lib file
49 F067 3D        DEC     A
50 F068 280F      JR      Z,DRDIS1      ; '???' file
51 F06A 3D        DEC     A
52 F06B 280C      JR      Z,DRDIS1      ; '???' file
53 F06D 1184F2    LD      DE,MSGQ10      ; 'SYS'
54 F070 3D        DEC     A
55 F071 2809      JR      Z,DRDIS0      ; Sys file
56 F073 1188F2    LD      DE,MSGQ11      ; 'GR '
57 F076 3D        DEC     A
58 F077 2803      JR      Z,DRDIS0      ; GR file
59 F079 118CF2    LD      DE,MSGQ??      ; '???'
60 F07C C5        ORDIS1: LD      DE,MSGQ??
        ORDIS0: PUSH    BC

```

```

01 F07D 0604          LD      B,04H
02 F07F CD0C00        DRDIS2: CALL PRNTS
03 F082 10FB          DJNZ   DRDIS2
04 F084 C1            POP     BC
05 F085 DF            RST     3
06 F086 CD0C00        CALL    PRNTS
07 F089 CD0C00        CALL    PRNTS
08 F08C CD0C00        CALL    PRNTS
09 F08F                ;
10 F08F                ;   Disp file name
11 F08F                ;
12 F08F 3E22          LD      A,'''
13 F091 CD1200        CALL    PRNT
14 F094 23            INC     HL
15 F095 E5            PUSH    HL
16 F096 01            POP     DE
17 F097 DF            RST     3
18 F098 3E22          LD      A,'''
19 F09A CD1200        CALL    PRNT
20 F09D CD0900        CALL    NL
21 F0A0                ;
22 F0A0                ;   Counter decrement
23 F0A0                ;
24 F0A0 111100        LD      DE,0011H
25 F0A3 19            ADD     HL,DE
26 F0A4 CDCA08        CALL    ?KEY
27 F0A7 B7            OR      A
28 F0A8 28FA          JR      Z,-4
29 F0AA CD320A        CALL    ?BRK
30 F0AD CA5EEA        JP      Z,ST1X
31 F0B0 108D          DJNZ   DSPATR      ; B=File counter
32 F0B2                ;
33 F0B2                ;   Directory end
34 F0B2                ;
35 F0B2 C395EF        QDOKM0: JP      QDOKM
36 F0B5                ;
37 F0B5                ;=====
38 F0B5                ;
39 F0B5                ;   Quick disk format command
40 F0B5                ;
41 F0B5                ;=====
42 F0B5                ;
43 F0B5                ;
44 F0B5                ;
45 F0B5                ;
46 F0B5                ;   Disp formatting
47 F0B5                ;
48 F0B5 1130F3        LD      DE,QDFMG      ; 'QD:FORMATTING'
49 F0B8 DF            RST     3
50 F0B9 CDCE00        CALL    WAITY      ; 'OK?(Y/N)'
51 F0BC                ;
52 F0BC                ;   Ready & wrpt check
53 F0BC                ;
54 F0BC CDE1EF        CALL    QDRWCK
55 F0BF 3808          JR      C,QER05
56 F0C1                ;
57 F0C1                ;   Format
58 F0C1                ;
59 F0C1 3E02          LD      A,02H      ; Init cmd set
60 F0C3 323011        LD      (QDPA),A

```

```

01 F0C6 CD10E1        CALL    QDIOS
02 F0C9 DA60F1        QER05: JP      C,QER06
03 F0CC                ;
04 F0CC                ;   Format end
05 F0CC                ;
06 F0CC 18E4          JR      QDOKM0
07 F0CE                ;
08 F0CE                ;   Wait key in 'Y'
09 F0CE                ;
10 F0CE CD0900        WAITY: CALL    NL
11 F0D1 1153F3        LD      DE,QDQCM0      ; 'OK?(Y/N)'
12 F0D4 DF            RST     3
13 F0D5 CD91F1        CALL    NKYWT      ; Wait non key
14 F0D8 CDB10F        WATKY: CALL    0FB1H      ; Call ?PONT
15 F0DB 3EEF          LD      A,EFH      ;
16 F0DD 77            LD      (HL),A
17 F0DE CDEBF0        CALL    WATIM
18 F0E1 3805          JR      C,WATK1      ; If 'Y' then CY=1
19 F0E3 7E            LD      A,(HL)
20 F0E4 EEEF          XOR     EFH
21 F0E6 18F5          JR      WATK0
22 F0E8 AF            WATK1: XOR     A
23 F0E9 77            LD      (HL),A
24 F0EA C9            RET
25 F0EB                ;
26 F0EB                ;   Wait timer
27 F0EB                ;
28 F0EB 010600        WATIM: LD      BC,0006H
29 F0EE 0B            WATIO: DEC     BC
30 F0EF 78            LD      A,B
31 F0F0 B1            OR      C
32 F0F1 C8            RET     Z
33 F0F2                ;
34 F0F2                ;   Key search
35 F0F2                ;
36 F0F2 3E01          LD      A,01H
37 F0F4 CD15F1        CALL    SRCK0
38 F0F7 FE7F          CP      7FH      ; 'Y' ?
39 F0F9 37            SCF
40 F0FA C8            RET     Z
41 F0FB 3E03          LD      A,03H
42 F0FD CD15F1        CALL    SRCK0
43 F100 FEFB          CP      FBH      ; 'N' ?
44 F102 2809          JR      Z,WAT11
45 F104 3E08          LD      A,08H
46 F106 CD15F1        CALL    SRCK0
47 F109 FE7E          CP      7EH      ; 'Break+Shift' ?
48 F10B 20E1          JR      NZ,WATIO
49 F10D                ;
50 F10D                ;   Jump ST1X
51 F10D                ;
52 F10D AF            WAT11: XOR     A
53 F10E 77            LD      (HL),A
54 F10F 31F010        LD      SP,10F0H
55 F112 C35EEA        JP      ST1X
56 F115                ;
57 F115                ;   Key scan
58 F115                ;
59 F115                ;
60 F115 E5            SRCK0: ENT     HL

```

```

01 F116 2100E0      LD      HL,E000H
02 F119 77          LD      (HL),A
03 F11A 23          INC     HL
04 F11B 7E          LD      A,(HL)
05 F11C F5          PUSH    AF
06 F11D C5          PUSH    BC
07 F11E 0614        LD      B,14H      ; 1ms*20=20ms
08 F120 CDB9E6      SRCK1: CALL    DLY1M      ; 1ms Delay
09 F123 10FB        DJNZ    SRCK1
10 F125 C1          POP     BC
11 F126 F1          POP     AF
12 F127 BE          CP      (HL)
13 F128 20F1        JR      NZ,SRCK2
14 F12A E1          POP     HL
15 F12B C9          RET
16 F12C             ;
17 F12C             ;=====
18 F12C             ;
19 F12C             ; Quick disk copy command
20 F12C             ;
21 F12C             ;=====
22 F12C             ;
23 F12C             ;
24 F12C CD27EF      QC:  ENT     QDRCK      ; Ready check
25 F12F 382F        JR      C,QR06
26 F131 CD8DEC      CALL    FNINP      ; Input file name
27 F134 3AA311      LD      A,(BUFF)
28 F137 FE0D        CP      0DH
29 F139 28F1        JR      Z,QC
30 F13B CD5FF2      CALL    HDPCL      ; Hard point clear
31 F13E 1191EE      LD      DE,MSGLD    ; 'Loading'
32 F141 DF          RST      3
33 F142             ;
34 F142             ; File search
35 F142             ;
36 F142 CDF7EE      CALL    FILSCH
37 F145 3819        JR      C,QR06
38 F147             ;
39 F147             ; Disp 'Loading ...'
40 F147             ;
41 F147 11F110      LD      DE,NAME
42 F14A DF          RST      3
43 F14B             ;
44 F14B             ; Iocs parameter set
45 F14B             ;
46 F14B 210012      LD      HL,1200H
47 F14E 223211      LD      (QDPC),HL
48 F151 2A0411      LD      HL,(QSIZE)
49 F154 223411      LD      (QDPE),HL
50 F157 210301      LD      HL,0103H    ; Read data block cmd.
51 F15A 223011      LD      (QDPA),HL
52 F15D             ;
53 F15D             ; Read data block
54 F15D             ;
55 F15D CD10E0      CALL    QDIOS
56 F160 3821        JR      C,QR07
57 F162             ;
58 F162             ; Bell
59 F162             ;
60 F162 CD3E00      CALL    BELL

```

```

01 F165             ;
02 F165             ; Disp 'Set Disk ...'
03 F165             ;
04 F165 CD0900      CALL    NL
05 F168 113EF3      LD      DE,QDQCM    ; 'Set destination disk'
06 F16B DF          RST      3
07 F16C             ;
08 F16C             ; Error return set
09 F16C             ;
10 F16C 3E02        QCSPT: LD      A,02H
11 F16E 323A11      LD      (QDCPA),A
12 F171 216CF1      LD      HL,QCSPT
13 F174 31EE10      LD      SP,10EEH
14 F177 E3          EX      (SP),HL
15 F178             ;
16 F178             ; Wait 'Y'
17 F178             ;
18 F178 CDCEFO      CALL    WAITY      ; 'OK?(Y/N)'
19 F17B             ;
20 F17B             ; Ready & wrpt check
21 F17B             ;
22 F17B CDE1EF      QCRWCK: CALL    QDRWCK
23 F17E 3803        JR      C,QR07
24 F180             ;
25 F180             ; File end search
26 F180             ;
27 F180 CD9FEF      CALL    FEDSCH      ; File end search
28 F183 DA02F2      JP      C,QRTRT    ; Already exist err
29 F186 FE28        CP      40          ; Not found ?
30 F188 C202F2      JP      NZ,QRTRT
31 F18B             ;
32 F18B             ; Save file start
33 F18B             ;
34 F18B 210012      LD      HL,1200H    ; Data adrs
35 F18E C374EF      JP      QSVIOS      ; Jump save routin
36 F191             ;
37 F191             ; Non key check
38 F191             ;
39 F191             ;
40 F191 060A        NKYWT: ENT
41 F193 2100E0      LD      B,0AH
42 F196 05          LD      HL,E000H
43 F197 70          DEC     B
44 F198 04          LD      (HL),B
45 F199 23          INC     B
46 F19A 7E          INC     HL
47 F19B FEFF        LD      A,(HL)
48 F19D 20F2        CP      FFH
49 F19F 10F2        JR      NZ,NKYWT
50 F1A1 C9          DJNZ    NKYWT0
51 F1A2             RET
52 F1A2             ;
53 F1A2             ;=====
54 F1A2             ;
55 F1A2             ; Quick disk xfer command
56 F1A2             ;
57 F1A2             ;=====
58 F1A2             ;
59 F1A2             ;
60 F1A2             ;

```

```

01 F1A2: ;
02 F1A2 C054EB ; CALL LD00 ; Read inf & data
03 F1A5 DA24EB JP C,?ERX
04 F1A8 09 EXX
05 F1A9 220411 LD (DTADR),HL
06 F1AC ;
07 F1AC ; Bell
08 F1AC ;
09 F1AC C03E00 CALL BELL
10 F1AF ;
11 F1AF ; Input file name
12 F1AF ;
13 F1AF C08DEC QXFN0: CALL FNINP ; Input file name
14 F1B2 3AA311 LD A,(BUFF)
15 F1B5 FE0D CP 0DH
16 F1B7 28F6 JR Z,QXFN0
17 F1B9 21A311 LD HL,BUFF
18 F1BC 11F110 LD DE,NAME
19 F1BF 011100 LD BC,0011H
20 F1C2 EDB0 LDIR
21 F1C4 ;
22 F1C4 ; Disp 'Set Disk ...'
23 F1C4 ;
24 F1C4 C00900 CALL NL
25 F1C7 113EF3 LD DE,QDQCM ; 'Set destination disk'
26 F1CA DF RST 3
27 F1CB ;
28 F1CB ; Atrb change (700800)
29 F1CB ;
30 F1CB LD A,(ATRB)
31 F1CE FE04 CP 04H ; 700 BSD file ?
32 F1D0 2806 JR Z,ATRC0
33 F1D2 FE05 CP 05H ; 700 BTX file ?
34 F1D4 2006 JR NZ,QXPRST
35 F1D6 3D DEC A
36 F1D7 3D DEC A
37 F1D8 3D DEC A
38 F1D9 32F010 ATRC0: DEC A ; Atrb set
39 F1DC LD (ATRB),A
40 F1DC ;
41 F1DC ; Parameter trans
42 F1DC 212E11 QXPRST: LD HL,112EH
43 F1DF 113011 LD DE,1130H
44 F1E2 012D00 LD BC,002DH
45 F1E5 EDB8 LDDR
46 F1E7 210000 LD HL,0000H ; Lock & Secret
47 F1EA 220211 LD (1102H),HL
48 F1ED ;
49 F1ED ; Error return set
50 F1ED ;
51 F1ED 3E02 QXSPST: LD A,02H
52 F1EF 323A11 LD (QDCPA),A
53 F1F2 21EDF1 LD HL,QXSPST
54 F1F5 31EE10 LD SP,10EEH
55 F1F8 E3 EX (SP),HL
56 F1F9 ;
57 F1F9 ; Wait 'Y'
58 F1F9 ;
59 F1F9 C0CEFO CALL WAITY ; 'OK?(Y/N)'
60 F1FC ;

```

```

01 F1FC ; Does flag reset
02 F1FC ;
03 F1FC C0ECEC CALL IOFRS
04 F1FF ;
05 F1FF ; Ready & wrpt check
06 F1FF ;
07 F1FF C37BF1 JP QCRWCK
08 F202 ;
09 F202 ;
10 F202 ;
11 F202 ; Error treatment
12 F202 ;
13 F202 ;
14 F202 ;
15 F202 ;
16 F202 1190F2 QERTRT: ENT LD DE,MGNFE ; 'Not Found err'
17 F205 FE28 CP NTFECO ; Not found err
18 F207 283B JR Z,QERMF
19 F209 1119F3 QERT0: LD DE,MGBDE ; 'Bad disk error'
20 F20C FE39 CP BDSKCD ; Bad disk err
21 F20E 2834 JR Z,QERMF
22 F210 11D6F2 QER1: LD DE,MGWPT ; 'Write protect'
23 F213 FE2E CP WPRTCD ; Write protect
24 F215 282D JR Z,QERMF
25 F217 11E7F2 QER2: LD DE,MGNRE ; 'Not ready'
26 F21A FE32 CP QNTRCD ; Not ready
27 F21C 2826 JR Z,QERMF
28 F21E 11F4F2 QER3: LD DE,MGNSE ; 'No file space error'
29 F221 FE35 CP NFSECD ; No file space err
30 F223 281F JR Z,QERMF
31 F225 1109F3 QER4: LD DE,MGUFE ; 'Unformat'
32 F228 FE36 CP UNFMCD ; Unformat err
33 F22A 2818 JR Z,QERMF
34 F22C 11C4F2 QER5: LD DE,MGALE ; 'Already exist'
35 F22F FE2A CP ALEXCD ; Already exist
36 F231 2811 JR Z,QERMF
37 F233 11A2F2 QER6: LD DE,MGTME ; 'Too many files'
38 F236 FE33 CP TMFECD ; Too many files err
39 F238 280A JR Z,QERMF
40 F23A 1129F3 QER7: LD DE,MGBRK ; 'Break!'
41 F23D FE00 CP BRKCD ; Break
42 F23F 2803 JR Z,QERMF
43 F241 11B8F2 QER8: LD DE,MGHDE ; 'Haard error'
44 F244 ;
45 F244 ; Motor off
46 F244 ;
47 F244 3E06 GERMF: LD A,06H ; Motor off cmd.
48 F246 323011 LD (QDPA),A
49 F249 C010E0 CALL QD10S
50 F24C C05FF2 CALL HDPCCL
51 F24F ;
52 F24F 3A3A11 LD A,(QDCPA)
53 F252 1F RRA
54 F253 D8 RET C ; Boot err
55 F254 F5 PUSH AF
56 F255 C00900 CALL NL
57 F258 DF RST 3
58 F259 F1 POP AF
59 F25A 1F RRA
60 F25B D8 RET C ; CMD QC,QX err

```

```

01 F25C C35EEA      JP      ST1X
02 F25F              ;
03 F25F              ; Header point clear
04 F25F              ;
05 F25F      HDPCL: ENT      LD      A,05H      ; Head point clear cmd.
06 F25F 3E05        LD      (QDPA),A
07 F261 323011      LD      CALL      QDIOS
08 F264 CD10E0      RET
09 F267 C9          ;
10 F268              ;=====
11 F268              ;
12 F268              ; Message table
13 F268              ;
14 F268              ;=====
15 F268              ;
16 F268      MSGQ0K: DEFM    'OK!'
17 F268 4F4B21      DEFB      0DH
18 F268 00          MSGQ01: DEFM    'OBJ'
19 F26C 4F424A      DEFB      0DH
20 F26F 00          MSGQ02: DEFM    'BTX'
21 F270 425458      DEFB      0DH
22 F273 00          MSGQ03: DEFM    'BSD'
23 F274 425344      DEFB      0DH
24 F277 00          MSGQ04: DEFM    'BRD'
25 F278 425244      DEFB      0DH
26 F27B 00          MSGQ05: DEFM    'RB '
27 F27C 524220      DEFB      0DH
28 F27F 00          MSGQ07: DEFM    'LIB'
29 F280 4C4942      DEFB      0DH
30 F283 00          MSGQ10: DEFM    'SYS'
31 F284 535953      DEFB      0DH
32 F287 00          MSGQ11: DEFM    'GR '
33 F288 475220      DEFB      0DH
34 F28B 00          MSGQ??: DEFM    '???'
35 F28C 3F3F3F      DEFB      0DH
36 F28F 00          MGNFE: DEFM    'QD:F'      ; 'QD:File not found'
37 F290 51443A46    DEFB      A6H      ; I
38 F294 A6          DEFB      B8H      ; L
39 F295 B8          DEFB      92H      ; E
40 F296 92          DEFB      20H      ;
41 F297 20          DEFB      B0H      ; N
42 F298 B0          DEFB      B7H      ; O
43 F299 B7          DEFB      96H      ; T
44 F29A 96          DEFB      20H      ;
45 F29B 20          DEFB      AAH      ; F
46 F29C AA          DEFB      B7H      ; O
47 F29D B7          DEFB      A5H      ; U
48 F29E A5          DEFB      B0H      ; N
49 F29F B0          DEFB      9CH      ; D
50 F2A0 9C          DEFB      0DH
51 F2A1 0D          ;
52 F2A2              ;
53 F2A2 51443A54    MGTME: DEFM    'QD:T'      ; 'Too many files err'
54 F2A6 B7          DEFB      B7H      ; O
55 F2A7 B7          DEFB      20H      ; O
56 F2A8 20          DEFB      B3H      ; M
57 F2A9 B3          DEFB      A1H      ; A
58 F2AA A1          DEFB      B0H      ; N
59 F2AB B0          DEFB      BDH      ; Y
60 F2AC BD

```

```

01 F2AD 20          DEFB      20H      ;
02 F2AE AA          DEFB      AAH      ; F
03 F2AF A6          DEFB      A6H      ; I
04 F2B0 B8          DEFB      B8H      ; L
05 F2B1 92          DEFB      92H      ; E
06 F2B2 A4          DEFB      A4H      ; S
07 F2B3 20          DEFB      20H      ;
08 F2B4 92          DEFB      92H      ; E
09 F2B5 9D          DEFB      9DH      ; R
10 F2B6 9D          DEFB      9DH      ; R
11 F2B7 0D          DEFB      0DH
12 F2B8              ;
13 F2B8 51443A48    MGHDE: DEFM    'QD:H'      ; 'QD:Hard err'
14 F2BC A1          DEFB      A1H      ; A
15 F2BD 9D          DEFB      9DH      ; R
16 F2BE 9C          DEFB      9CH      ; D
17 F2BF 20          DEFB      20H      ;
18 F2C0 92          DEFB      92H      ; E
19 F2C1 9D          DEFB      9DH      ; R
20 F2C2 9D          DEFB      9DH      ; R
21 F2C3 0D          DEFB      0DH
22 F2C4 41          MGALE: DEFM    'A'        ; 'Already exist err'
23 F2C5 B8          DEFB      B8H      ; L
24 F2C6 9D          DEFB      9DH      ; R
25 F2C7 92          DEFB      92H      ; E
26 F2C8 A1          DEFB      A1H      ; A
27 F2C9 9C          DEFB      9CH      ; D
28 F2CA BD          DEFB      BDH      ; Y
29 F2CB 20          DEFB      20H      ;
30 F2CC 92          DEFB      92H      ; E
31 F2CD 9B          DEFB      9BH      ; X
32 F2CE A6          DEFB      A6H      ; I
33 F2CF A4          DEFB      A4H      ; S
34 F2D0 96          DEFB      96H      ; T
35 F2D1 20          DEFB      20H      ;
36 F2D2 92          DEFB      92H      ; E
37 F2D3 9D          DEFB      9DH      ; R
38 F2D4 9D          DEFB      9DH      ; R
39 F2D5 0D          DEFB      0DH
40 F2D6              ;
41 F2D6 51443A57    MGWPT: DEFM    'QD:W'      ; 'QD:Write protect'
42 F2DA 9D          DEFB      9DH      ; R
43 F2DB A6          DEFB      A6H      ; I
44 F2DC 96          DEFB      96H      ; T
45 F2DD 92          DEFB      92H      ; E
46 F2DE 20          DEFB      20H      ;
47 F2DF 9E          DEFB      9EH      ; P
48 F2E0 9D          DEFB      9DH      ; R
49 F2E1 B7          DEFB      B7H      ; O
50 F2E2 96          DEFB      96H      ; T
51 F2E3 92          DEFB      92H      ; E
52 F2E4 9F          DEFB      9FH      ; C
53 F2E5 96          DEFB      96H      ; T
54 F2E6 0D          DEFB      0DH
55 F2E7              ;
56 F2E7 51443A4E    MGNRE: DEFM    'QD:N'      ; 'QD:Not ready'
57 F2EB B7          DEFB      B7H      ; O
58 F2EC 96          DEFB      96H      ; T
59 F2ED 20          DEFB      20H      ;
60 F2EE 9D          DEFB      9DH      ;

```

```

01 F2EF 92      DEFB 92H      ; E
02 F2F0 A1      DEFB A1H      ; A
03 F2F1 9C      DEFB 9CH      ; D
04 F2F2 8D      DEFB 8DH      ; Y
05 F2F3 0D      DEFB 0DH
06 F2F4
07 F2F4 51443A4E  MGNSE: DEFM 'QD:N'      ; 'QD:No file space err'
08 F2F8 B7      DEFB B7H      ; O
09 F2F9 20      DEFB 20H      ;
10 F2FA AA      DEFB AAH      ; F
11 F2FB A6      DEFB A6H      ; I
12 F2FC B8      DEFB B8H      ; L
13 F2FD 92      DEFB 92H      ; E
14 F2FE 20      DEFB 20H      ;
15 F2FF A4      DEFB A4H      ; S
16 F300 9E      DEFB 9EH      ; P
17 F301 A1      DEFB A1H      ; A
18 F302 9F      DEFB 9FH      ; C
19 F303 92      DEFB 92H      ; E
20 F304 20      DEFB 20H      ;
21 F305 92      DEFB 92H      ; E
22 F306 9D      DEFB 9DH      ; R
23 F307 9D      DEFB 9DH      ; R
24 F308 0D      DEFB 0DH
25 F309
26 F309 51443A55  MGUFE: DEFM 'QD:U'      ; 'QD:Unformat err'
27 F30D B0      DEFB B0H      ; N
28 F30E AA      DEFB AAH      ; F
29 F30F B7      DEFB B7H      ; O
30 F310 9D      DEFB 9DH      ; R
31 F311 B3      DEFB B3H      ; M
32 F312 A1      DEFB A1H      ; A
33 F313 96      DEFB 96H      ; T
34 F314 20      DEFB 20H      ;
35 F315 92      DEFB 92H      ; E
36 F316 9D      DEFB 9DH      ; R
37 F317 9D      DEFB 9DH      ; R
38 F318 0D      DEFB 0DH
39 F319
40 F319 51443A42  MGBDE: DEFM 'QD:B'      ; 'QD:Bad disk err'
41 F31D A1      DEFB A1H      ; A
42 F31E 9C      DEFB 9CH      ; D
43 F31F 20      DEFB 20H      ;
44 F320 9C      DEFB 9CH      ; D
45 F321 A6      DEFB A6H      ; I
46 F322 A4      DEFB A4H      ; S
47 F323 A9      DEFB A9H      ; K
48 F324 20      DEFB 20H      ;
49 F325 92      DEFB 92H      ; E
50 F326 9D      DEFB 9DH      ; R
51 F327 9D      DEFB 9DH      ; R
52 F328 0D      DEFB 0DH
53 F329
54 F329          ;
55 F329 42      MGBRK: ENT DEFM 'B'      ; 'Break!'
56 F32A 9D      DEFB 9DH      ; R
57 F32B 92      DEFB 92H      ; E
58 F32C A1      DEFB A1H      ; A
59 F32D A9      DEFB A9H      ; K
60 F32E 21      DEFB 21H      ; !

```

```

01 F32F 0D      DEFB 0DH
02 F330 51443A46  GDIMG: DEFM 'QD:F'      ; 'QD:Formatting'
03 F334 B7      DEFB B7H      ; O
04 F335 9D      DEFB 9DH      ; R
05 F336 B3      DEFB B3H      ; M
06 F337 A1      DEFB A1H      ; A
07 F338 96      DEFB 96H      ; T
08 F339 96      DEFB 96H      ; T
09 F33A A6      DEFB A6H      ; I
10 F33B B0      DEFB B0H      ; N
11 F33C 97      DEFB 97H      ; G
12 F33D 0D      DEFB 0DH
13 F33E
14 F33E 53      ;
15 F33F 92      GDQCM: DEFM 'S'      ; 'Set destination disk'
16 F340 96      DEFB 96H      ; E
17 F341 20      DEFB 20H      ; T
18 F342 9C      DEFB 9CH      ;
19 F343 92      DEFB 92H      ; D
20 F344 A4      DEFB A4H      ; E
21 F345 96      DEFB 96H      ; S
22 F346 A6      DEFB A6H      ; T
23 F347 B0      DEFB B0H      ; I
24 F348 A1      DEFB A1H      ; N
25 F349 96      DEFB 96H      ; A
26 F34A A6      DEFB A6H      ; T
27 F34B B7      DEFB B7H      ; I
28 F34C B0      DEFB B0H      ; O
29 F34D 20      DEFB 20H      ;
30 F34E 9C      DEFB 9CH      ; D
31 F34F A6      DEFB A6H      ;
32 F350 A4      DEFB A4H      ; I
33 F351 A9      DEFB A9H      ; S
34 F352 0D      DEFB 0DH      ; K
35 F353
36 F353 4F4B3F28  ;
37 F357 592F4E29  GDQCM0: DEFM 'OK?(Y/N)'
38 F35B 0D      DEFB 0DH
39 F35C 44      DIRMSG: DEFM 'D'      ; 'Directory of QD:'
40 F35D A6      DEFB A6H      ; I
41 F35E 9D      DEFB 9DH      ; R
42 F35F 92      DEFB 92H      ; E
43 F360 9F      DEFB 9FH      ; C
44 F361 96      DEFB 96H      ; T
45 F362 B7      DEFB B7H      ; O
46 F363 9D      DEFB 9DH      ; R
47 F364 BD      DEFB BDH      ; Y
48 F365 20      DEFB 20H      ;
49 F366 B7      DEFB B7H      ; O
50 F367 AA      DEFB AAH      ; F
51 F368 20      DEFB 20H      ;
52 F369 51      DEFB 51H      ; Q
53 F36A 44      DEFB 44H      ; D
54 F36B 3A      DEFB 3AH      ;
55 F36C 0D      DEFB 0DH
56 F36D
57 F36D          ; SKP H

```

```

01 F360      ;
02 F360      ; Quick disk lacs (Synes)
03 F360      ; Patch
04 F360      ; 1984.10.11
05 F360      ;
06 F380      ;
07 F380      EOMRS:  ORG    F380H
08 F380 3EC0      LD      A,C0H      ; Eom Reset
09 F382 D3F6      OUT     (SIOAC),A
10 F384 C9        RET
11 F385      ;
12 F385      END

```

```

#CLR8 09D5 .4DE 02A6 12ERMG EE3A 12NSET E7DF 1R12R0 E6DA
1R12R0 E6CA 1R12WR E745 2HEX 041F 70M00 E008 ?ADCN 00B9
?BEL 0577 ?BLNK 00A6 ?BRK 0A32 ?CLER 0F08 ?DPCT 00DC
?ERX EB24 ?ERX0 EB2E ?KEY 08CA ?MODE 073E ?PRTS 0920
?RDD 04F8 ?RDI 0408 ?TMST 0308 ?VRFY 0588 ?VRFY0 0593
?WRD 0475 ?WR1 0436 ADJ E63C AINP0 ECE0 ALEXCD 002A
ASC 03DA ATRB 10F0 ATRC0 F108 BDR E0FF BDSKCD 0039
BDV E21B BDV1 E220 BDW E1ED BDW1 E204 BEGIN E813
BEGIN0 E800 BELL 003E BFRF 1146 BGETL 012F BGETLX EB34
BG10 E876 BG1N00 E804 BLKTR ED20 BOOT E4D1 BPARA CEE9
BPRO CF00 BREAD E5A7 BRKC E435 BRKCD 0000 BRKCK 001E
BRKEY 001E BSDCD 0003 BSY0 E5A1 BSYOF0 E58F BSYOF1 E59C
BSYOF2 E58C BSYOF3 E597 BSYOF4 E59F BSYOFF E587 BSYON E568
BSYON0 E570 BSYON1 E57D BSYON2 E56D BSYON3 E578 BSYONE E580
BTXCD 0002 BUF 11A3 BUFR 11A3 BUFF 11A3 CAL0 EAA0
CHECK E729 CK1 E730 CK2 E735 CLBF0 CEF6 CLBFI CEF7
CLBF2 CEF8 CLBF3 CEF9 CLEET EA43 CMD CEF4 CM00T1 E555
CM00T2 E64E CMY0 0058 CMY00 E85F CNVRT E696 COMNT 1108
CR 0008 CSS E945 DIREFC F023 DIRIOP F003 DIRMSG F35C
DIRMTF F02B DISCLR EA59 DLY1M E6B9 DLY60M E6BF DLY80U E6B3
DLYT E6C3 DM 00DC DMR E132 DMR1 E13A DR 000B
DRDIS0 F07C DRDIS1 F079 DRDIS2 F07F DSFLNA EEC2 DSILF E972
DSPATR F03F DSPXY 1171 DTADR 1104 DUM1X EC87 DUM2X EC3E
DUM3X EC37 DUMPX EC29 EAINP ECD8 EDFLCK EFB8 EOM E3B2
EOMRS F380 ERCK1 E683 ERJMP E690 ERRET E68E ERRET1 E68A
ERRM0 EDE5 ERRM1 EDF6 ERRM2 EE04 ERRM3 EE16 ERRM4 EE27
ERRMG EA34 ERROR E661 ERRTR0 E4BF ERRTR1 E4C2 ERRTRT E4B6
EXADR 1106 EXCMT E990 EXF ECFC EXF0 ED40 EXRAM E6D0
FDB E8CC FDCC& E805 FDOCK EB00 FDMTON E517 FDX E44A
FEDSCH EF9F FILESH EEB6 FILSCH EEF7 FMERR E006 FMS EFFF
FNA 1142 FNB 1143 FNEND E2CE FNINP EC8D FNINP4 ECC5 FNINP5 ECCF
FNINP1 ECB1 FNINP2 ECB9 FNINP3 ECBD FNINP4 ECC5 FNINP5 ECCF
FNINP6 ECD1 FNINPR EC9F FNUPF 1141 FNUPS 113F FNUPS1 1140
GDGM0 00CE GDGRF 00C0 GDGW F 00CC GETKY 001B GETL 0003
GOT0 EAAC GOTO 00F3 GOTOX EAA9 HDERC0 0029 HDPC L F25F
HDPT 113D HDPT0 113E HEXIYX EB42 HLHEX 0410 HPNFE E14A
HPS E114 HPS1 E147 HS 00DD INTER E658 IOE41 E400
IOE46 E403 IOE50 E406 IOE53 E409 IOE54 E40C IOFRS EEEC
IPLM0 ED88 IPLM1 ED98 IPLM2 EDA7 IPLM3 EDB5 IPLM4 EDC3
IPLMC E4CA JST1X EAE2 KEYCK E8B8 KEYPA E000 KEYPB E001
KSCAN0 E90F KSCAN1 E91E KSCAN2 E932 KSJPOB E91B LCM12 E807
LD00 EB54 LDCMT EB70 LDDEC E02E LDFNCK EF1B LDINC ED3E
LET EA48 LETNL 0006 LLPT 0470 LOA0X EB58 LOADX EB4C
LPARA0 EECB LPARA1 EECE LREDY E2A3 LSINT E2D8 MASTE E4AE
MCHECK E471 MCORX EB7B MCR1X EB7E MCR2X EBA7 MCR3X EBAA
MGAL E2C4 MGBDE F319 MGBRK F329 MGHDE F28B MGNFE F290
MGNRE F2E7 MGNSE F2F4 MGTME F2A2 MGUF E309 MGWPT F2D6
MLDSP 02BE MOFF E530 MON E93B MONMSG EE4D MSG 0015
MSG?3 06E7 MSGE1 0147 MSGEA EE7B MSGLO EE91 MSGLET EA4E
MSGOK 0942 MSGPS EA50 MSGQ01 F26C MSGQ02 F270 MSGQ03 F274
MSGQ04 F278 MSGQ05 F27C MSGQ07 F280 MSGQ10 F284 MSGQ11 F288
MSGQ?? F28C MSGQOK F268 MSGSV0 EE65 MSGTA EE70 MSGW01 E7E2
MSGW02 E7F0 MSGXA EE86 MT01 E51D MTF 1144 MTFG CEF5
MTOF E2E8 MTOFX E003 MTON E29B NAME 10F1 NAMSIZ 0011
NFNUP E1E8 NFSECD 0035 NKYWT F191 NKYWT0 F193 NL 0009
NLPHL 05FA NOTIO E4BA NRCK E3E5 NTFEC0 0028 OBJCD 0001
PARST1 E5FB PARST2 E62B P1OA E835 P1OIDA EE9A P1OIDB EE9E
PLTDT EEA2 PLTST E8E1 PRNT 0012 PRNT3 096C PRNTS 000C
PRTHL 03BA PS0 E862 PSGET E866 QBFLSH E9E0 QBT E9B7
QC F12C QCRWCK F17B QCSPST F16C QD EFEF QOATRC EE8B

```

QDCPA	113A	QDCPB	113B	QDCPC	113C	QDFM	E090	QDFM1	E09C
QDFM2	E0A9	QDFM3	E0C0	QDFM4	E0D0	QDFMER	E9FA	QDFMG	F330
QDHPC	E083	QDIO	E810	QDIOS	E010	QDIRBF	CD90	QDLDER	EA04
QDNTR	E9F7	QDOFF	E2E8	QDOKM	EF95	QDOKM0	F0B2	QDPA	1130
QDPB	1131	QDPC	1132	QDPE	1134	QDPG	1136	QDPI	1138
QDQCM	F33E	QDQCM0	F353	QDRC	E08A	QDRCK	EF27	QDRD	E0DA
QDRWC0	EFE6	QDRWCK	EFE1	QDSVFN	E268	QDTADR	1106	QDTBL	1130
QDWE1	E1E5	QDWR	E14E	QDWR1	E171	QDWR2	E185	QDWR3	E1CE
QDWR4	E1AC	QDWR5	E1E2	QER00	EEEE	QER01	EF31	QER03	EF92
QER04	F028	QER05	F0C9	QER06	F160	QER07	F183	QER1	F210
QER2	F217	QER3	F21E	QER4	F225	QER5	F22C	QER6	F233
QER7	F23A	QER8	F241	QERMF	F244	QERT0	F209	QERTRT	F202
QEXADR	1108	QF	F0B5	QL	EEA7	QLINF	EF09	QMEIN	E06A
QNTRCD	0032	QREDY	E23C	QS	EF2E	QSIOST	EF71	QSIZE	1104
QSVIOS	EF74	QUICK	EAB5	QX	F1A2	QXFNO	F1AF	QXPRST	F10C
QXSPST	F1E0	R12C&B	E8F2	R12CK1	E7BA	RAMBRD	EAE5	RCLB	E548
RDATA	E3F0	RDB0	E769	RDCR1	E3C5	RDCR2	E3CA	RDCRC	E3C3
RDDAT	002A	RDDTBL	EB6D	RDINF	0027	ROTB0	E997	RE3	E5E0
RE4	E5C1	RE6	E5BF	RE8	E5AD	READ1	E5AA	READY	E4DC
REDY0	E4F4	REDY1	E512	REDY2	E511	REND	E5F6	RETSP	1148
RMBD1	EAF3	RSET	E211	RTY	E015	RTY2	E062	RTY3	E045
RTY4	E068	RTY5	E04D	RTYF	1145	SAC00	EBF3	SAC01	EBF8
SACMT	EBE8	SAVEX	EBAE	SAVJE	EBDF	SAVJP	EBE5	SBRK	E39A
SCM12	E80A	SCR	00DA	SEEK	E528	SELM0	ED43	SELM00	EA1D
SELM01	EA28	SELM1	ED53	SELM2	ED61	SELM3	ED6E	SELM4	ED7E
SELMG0	E909	SELMG	EA06	SFNC	EFD1	SFNC0	EFDD	SGJP	EC27
SGX	EC1E	SIDST	E61B	SIOAC	00F6	SIOAD	00F4	SIOBC	00F7
SIOBD	00F5	SIOCK	EB13	SIOLO	E421	SIOSD	E42C	SIZE	1102
SJP	E462	SLMSG	E906	SLPT	03D5	SORES	E86C	SP	10F0
SPHEX	03B1	SRCK0	F115	SRCK1	F120	SRCK2	F11B	SREDY	E271
ST1X	EA5E	ST2X	EA6F	STAFG	CEFB	SUM	E70E	SUMCK1	E715
SUMCK2	E727	SUMCK3	E71C	SVFLST	EF6C	SWRK	119D	SYNCA	E3A3
SYNCF	1147	SYNCL1	E313	SYNCL2	E2FD	SYNCS1	E37E	SYNCS2	E365
SYNCW0	E32C	SYNCW1	E33D	SYNCW2	E347	SYNCW3	E357	SYNW01	E354
TAINP	EC03	TEMPW	119E	TIMIN	038D	TIMW	E414	TIMW1	E415
TIMW2	E417	TMFCD	0033	TMLPL	E316	TMLPS	E381	TPERJP	E9B4
TPERX	E9AA	TPERX0	E9AC	TR	00D9	TRA1	E3DC	TRANS	E3DB
TRANS0	E6A2	TRANS1	E6A9	UNFMCD	0036	VCM12	E80D	VFCMT	EC09
VRF120	EC10	VRFCNT	CEFA	VRFYE	EC03	VRFYX	EC00	WAITY	F0CE
WATI0	F0EE	WATI1	F10D	WATIM	F0EB	WATK0	F0DD	WATK1	F0E8
WATKY	F0D8	WCK1	E796	WCK2	E79A	WCK3	E79E	WMOD0	E948
WMOD1	E968	WPRTCD	002E	WRCAN	E1DB	XAINP	ECDD		

4 Z-80 Programming reference

4-1 Z-80 Status indicators (Flags)

The flag register (F and F') supplies information to the user regarding the status of the Z80 at any given time. The bit positions for each flag is shown below:

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

where:

- C = Carry Flag
- N = Add/Subtract Flag
- P/V = Parity/Overflow Flag
- H = Half-Carry Flag
- Z = Zero Flag
- S = Sign Flag
- X = Not Used

Each of the two Z-80 Flag Registers contains 6 bits of status information which are set or reset by CPU operations. (Bits 3 and 5 are not used.) Four of these bits are testable (C, P/V, Z and S) for use with conditional jump, call or return instructions. Two flags are not testable (H, N) and are used for BCD arithmetic.

Carry Flag (C)

The carry bit is set or reset depending on the operation being performed. For 'ADD' instructions that generate a carry and 'SUBTRACT' instructions that generate no borrow, the Carry Flag will be set. The Carry Flag is reset by an ADD that does not generate a carry and a 'SUBTRACT' that generates a borrow. This saved carry facilitates software routines for extended precision arithmetic. Also, the 'DAA' instruction will set the Carry Flag if the conditions for making the decimal adjustment are met.

For instructions RLA, RRA, RLS and RRS, the carry bit is used as a link between the LSB and MSB for any register or memory location. During instructions RLCA, RLCA s and SLA s, the carry contains the last value shifted out of bit 7 of any register or memory location. During instructions RRCA, RRC s, SRA s and SRL s, the carry contains the last value shifted out of bit 0 of any register or memory location.

For the logical instructions AND s, OR s and XOR s, the carry will be reset.

The Carry Flag can also be set (SCF) and complemented (CCF).

Add/Subtract Flag (N)

This flag is used by the decimal adjust accumulator instruction (DAA) to distinguish between 'ADD' and 'SUBTRACT' instructions. For all 'ADD' instructions, N will be set to an '0'. For all 'SUBTRACT' instructions, N will be set to a '1'.

The Zero Flag (Z)

The Zero Flag (Z) is set or reset if the result generated by the execution of certain instructions is a zero.

For 8-bit arithmetic and logical operations, the Z flag will be set to a '1' if the resulting byte in the Accumulator is zero. If the byte is not zero, the Z flag is reset to '0'.

For compare (search) instructions, the Z flag will be set to a '1' if a comparison is found between the value in the Accumulator and the memory location pointed to by the contents of the register pair HL.

When testing a bit in a register or memory location, the Z flag will contain the complemented state of the indicated bit (see Bit b,s).

When inputting or outputting a byte between a memory location and an I/O device (INI, IND, OUTI and OUTD) if the result of B-1 is zero, the Z flag is set, otherwise it is reset. Also for byte inputs from I/O devices using IN r, (C), the Z flag is set to indicate a zero byte input.

Parity/Overflow Flag

This flag is set to a particular state depending on the operation being performed.

For arithmetic operations, this flag indicates an overflow condition when the result in the Accumulator is greater than the maximum possible number (+127) or is less than the minimum possible number (−128). This overflow condition can be determined by examining the sign bits of the operands.

For addition, operands with different signs will never cause overflow. When adding operands with like signs and the result has a different sign, the overflow flag is set. For example:

+120 = 0111 1000	ADDEND
+105 = 0110 1001	AUGEND
+225 1110 0001	(−95) SUM

The two numbers added together has resulted in a number that exceeds +127 and the two positive operands has resulted in a negative number (−95) which is incorrect. The overflow flag is therefore set.

For subtraction, overflow can occur for operands of unlike signs. Operands of like sign will never cause overflow. For example:

+127 0111 1111	MINUEND
(−) −64 1100 0000	SUBTRAHEND
+191 1011 1111	DIFFERENCE

The minuend sign has changed from a positive to a negative, giving an incorrect difference. Overflow is therefore set.

Another method for predicting an overflow is to observe the carry into and out of the sign bit. If there is a carry in and no carry out, or if there is no carry in and a carry out, then overflow has occurred.

This flag is also used with logical operations and rotate instructions to indicate the parity of the result. The number of '1' bits in a byte are counted. If the total is odd, 'ODD' parity (P=0) is flagged. If the total is even, 'EVEN' parity is flagged (P=1).

During search instructions (CPI, CPIR, CPD, CPDR) and block transfer instructions (LDI, LDIR, LDD, LDDR) the P/V flag monitors the state of the byte count register (BC). When decrementing, the byte counter results in a zero value, the flag is reset to 0, otherwise the flag is a Logic 1.

During LD A, I and LD A, R instructions, the P/V flag will be set with the contents of the interrupt enable flip-flop (1FF2) for storage or testing.

When inputting a byte from an I/O device, IN r, (C), the flag will be adjusted to indicate the parity of the data.

The Half Carry Flag (H)

The Half Carry Flag (H) will be set or reset depending on the carry and borrow status between bits 3 and 4 of an 8-bit arithmetic operation. This flag is used by the decimal adjust accumulator instruction (DAA) to correct the result of a packed BCD add or subtract operation. The H flag will be set (1) or reset (0) according to the following table:

H	ADD	SUBTRACT
1	There is a carry from Bit 3 to Bit 4.	There is no borrow from bit 4.
0	There is no carry from Bit 3 to Bit 4.	There is a borrow from Bit 4.

The Sign Flag (S)

The Sign Flag (S) stores the state of the most significant bit of the Accumulator (Bit 7). When the Z80 performs arithmetic operations on signed numbers, binary two's complement notation is used to represent and process numeric information. A positive number is identified by a '0' in bit 7. A negative number is identified by a '1'. The binary equivalent of the magnitude of a positive number is stored in bits 0 to 6 for a total range of from 0 to 127. A negative number is represented by the two's complement of the equivalent positive number. The total range for negative numbers is from −1 to −128.

When inputting a byte from an I/O device to a register, IN r, (C), the S flag will indicate either positive (S=0) or negative (S=1) data.

4-2 Notation

instruction	flags						comments
	C	Z	P/V	S	N	H	
ADD A, s; ADC A, s	:	:	V	:	0	:	8-bit add or add with carry
SUB s; SBC A, s, CP s, NEG	:	:	V	:	1	:	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	0	:	P	:	0	1	Logical operations
OR s; XOR s	0	:	P	:	0	0	And sets different flags
INC s	●	:	V	:	0	:	8-bit increment
DEC m	●	:	V	:	1	:	8-bit decrement
ADD DD, ss	:	●	●	●	0	X	16-bit add
ADC HL, ss	:	:	V	:	0	X	16-bit add with carry
SBC HL, ss	:	:	V	:	1	X	16-bit subtract with carry
RLA; RLCA, RRA, RRCA	:	●	●	●	0	0	Rotate accumulator
RL m; RLC m; RR m; RRC m SLA m; SRA m; SRL m	:	:	P	:	0	0	Rotate and shift locations
RLD, RRD	●	:	P	:	0	0	Rotate digit left and right
DAA	:	:	P	:	●	:	Decimal adjust accumulator
CPL	●	●	●	●	1	1	Complement accumulator
SCF	1	●	●	●	0	0	Set carry
CCF	:	●	●	●	0	X	Complement carry
IN r, (C)	●	:	P	:	0	0	Input register indirect
INI; IND; OUTI; OUTD	●	:	X	X	1	X	Block input and output Z=0 if B≠0 otherwise Z=1
INIR; INDR; OTIR; OTDR	●	1	X	X	1	X	
LDI, LDD	●	X	:	X	0	0	Block transfer instructions P/V=1 if BC≠0, otherwise P/V=0
LDIR, LDDR	●	X	0	X	0	0	
CPI, CPIR, CPD, CPDR	●	:	:	X	1	X	Block search instructions Z=1 if A=(HL), otherwise Z=0 P/V=1 if BC≠0, otherwise P/V=0
LD A, I; LD A, R	●	:	IFF	:	0	0	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
BIT b, s	●	:	X	X	0	1	The complement of bit b of location is copied into the Z flag
NEG	:	:	V	:	1	:	Negate accumulator

The following notation is used in this table:

SYMBOL

OPERATION

- C Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
- Z Zero flag. Z=1 if the result of the operation is zero.
- S Sign flag. S=1 if the MSB of the result is one.
- P/V Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
- H Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.
- N Add/Subtract flag. N=1 if the previous operation was a subtract.
- H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
- :
- The flag is affected according to the result of the operation.
-
- The flag is unchanged by the operation.
- 0
- The flag is reset by the operation.
- 1
- The flag is set by the operation.
- X
- The flag is a "don't care."
- V
- P/V flag affected according to the overflow result of the operation.
- P
- P/V flag affected according to the parity result of the operation.
- r
- Any one of the CPU registers A, B, C, D, E, H, L.
- s
- Any 8-bit location for all the addressing modes allowed for the particular instruction.
- ss
- Any 16-bit location for all the addressing modes allowed for that instruction.
- ii
- Any one of the two index registers IX or IY.
- R
- Refresh counter.
- n
- 8-bit value in range <0,255>.
- nn
- 16-bit value in range <0,65535>.
- m
- Any 8-bit location for all the addressing modes allowed for the particular instruction.

8-Bit Load Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
LD r,r'	$r \leftarrow r'$	●	●	●	●	●	●	●	01	r	r'	1	1	4	r, r
LD r,n	$r \leftarrow n$	●	●	●	●	●	●	●	00	r	110	2	2	7	000 B
										$\leftarrow n \rightarrow$					001 C
LD r,(HL)	$r \leftarrow (HL)$	●	●	●	●	●	●	●	01	r	110	1	2	7	010 D
LD r,(IX+d)	$r \leftarrow (IX+d)$	●	●	●	●	●	●	●	11	011	101	3	5	19	011 E
									01	r	110				100 H
										$\leftarrow d \rightarrow$					101 L
LD r,(IY+d)	$r \leftarrow (IY+d)$	●	●	●	●	●	●	●	11	111	101	3	5	19	111 A
									01	r	110				
										$\leftarrow d \rightarrow$					
LD (HL),r	$(HL) \leftarrow r$	●	●	●	●	●	●	●	01	110	r	1	2	7	
LD (IX+d),r	$(IX+d) \leftarrow r$	●	●	●	●	●	●	●	11	011	101	3	5	19	
									01	110	r				
										$\leftarrow d \rightarrow$					
LD (IY+d),r	$(IY+d) \leftarrow r$	●	●	●	●	●	●	●	11	111	101	3	5	19	
									01	110	r				
										$\leftarrow d \rightarrow$					
LD (HL),n	$(HL) \leftarrow n$	●	●	●	●	●	●	●	00	110	110	2	3	10	
										$\leftarrow n \rightarrow$					
LD (IX+d),n	$(IX+d) \leftarrow n$	●	●	●	●	●	●	●	11	011	101	4	5	19	
									00	110	110				
										$\leftarrow d \rightarrow$					
										$\leftarrow n \rightarrow$					
LD (IY+d),n	$(IY+d) \leftarrow n$	●	●	●	●	●	●	●	11	111	101	4	5	19	
									00	110	110				
										$\leftarrow d \rightarrow$					
										$\leftarrow n \rightarrow$					
LD A,(BC)	$A \leftarrow (BC)$	●	●	●	●	●	●	●	00	001	010	1	2	7	
LD A,(DE)	$A \leftarrow (DE)$	●	●	●	●	●	●	●	00	011	010	1	2	7	
LD A,(nn)	$A \leftarrow (nn)$	●	●	●	●	●	●	●	00	111	010	3	4	13	
										$\leftarrow n \rightarrow$					
										$\leftarrow n \rightarrow$					
LD (BC),A	$(BC) \leftarrow A$	●	●	●	●	●	●	●	00	000	010	1	2	7	
LD (DE),A	$(DE) \leftarrow A$	●	●	●	●	●	●	●	00	010	010	1	2	7	
LD (nn),A	$(nn) \leftarrow A$	●	●	●	●	●	●	●	00	110	010	3	4	13	
										$\leftarrow n \rightarrow$					
										$\leftarrow n \rightarrow$					
LD A,I	$A \leftarrow I$	●	↑	IFF2	↑	0	0	0	11	101	111	2	2	9	IFF2: contents of interrupt enable flip-flop 2
									01	010	111				
LD A,R	$A \leftarrow R$	●	↑	IFF2	↑	0	0	0	11	101	101	2	2	9	
									01	011	111				
LD I,A	$I \leftarrow A$	●	●	●	●	●	●	●	11	101	101	2	2	9	
									01	000	111				
LD R,A	$R \leftarrow A$	●	●	●	●	●	●	●	11	101	101	2	2	9	
									01	001	111				

16-Bit Load Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
LD dd,nn	dd ← nn	•	•	•	•	•	•	•	00	dd0	001	3	3	10	dd register pairs
									←	n	→				00 BC
									←	n	→				01 DE
LD IX,nn	IX ← nn	•	•	•	•	•	•	•	11	011	101	4	4	14	10 HL
									00	100	001				11 SP
									←	n	→				
LD IY,nn	IY ← nn	•	•	•	•	•	•	•	11	111	101	4	4	14	
									00	100	001				
									←	n	→				
LD HL,(nn)	H ← (nn + 1) L ← (nn)	•	•	•	•	•	•	•	00	101	010	3	5	16	
									←	n	→				
									←	n	→				
LD dd,(nn)	dd _H ← (nn + 1) dd _L ← (nn)	•	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd1	011				
									←	n	→				
LD IX,(nn)	IX _H ← (nn + 1) IX _L ← (nn)	•	•	•	•	•	•	•	11	011	101	4	6	20	
									00	101	010				
									←	n	→				
LD IY,(nn)	IY _H ← (nn + 1) IY _L ← (nn)	•	•	•	•	•	•	•	11	111	101	4	6	20	
									00	101	010				
									←	n	→				
LD (nn),HL	(nn + 1) ← H (nn) ← L	•	•	•	•	•	•	•	00	100	010	3	5	16	
									←	n	→				
									←	n	→				
LD (nn),dd	(nn + 1) ← dd _H (nn) ← dd _L	•	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd0	011				
									←	n	→				
LD (nn),IX	(nn + 1) ← IX _H (nn) ← IX _L	•	•	•	•	•	•	•	11	011	101	4	6	20	
									00	100	010				
									←	n	→				
LD (nn),IY	(nn + 1) ← IY _H (nn) ← IY _L	•	•	•	•	•	•	•	11	111	101	4	6	20	
									00	100	010				
									←	n	→				
LD SP,HL	SP ← HL	•	•	•	•	•	•	•	11	111	001	1	1	6	
LD SP,IX	SP ← IX	•	•	•	•	•	•	•	11	011	101	2	2	10	
LD SP,IY	SP ← IY	•	•	•	•	•	•	•	11	111	101	2	2	10	
									11	111	001				

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
PUSH qq	$(SP-2) \leftarrow qq_L$	●	●	●	●	●	●	●	11	qq0	101	1	3	11	qq
	$(SP-1) \leftarrow qq_H$														register pairs
PUSH IX	$(SP-2) \leftarrow IX_L$	●	●	●	●	●	●	●	11	011	101	2	4	15	00
	$(SP-1) \leftarrow IX_H$								11	100	101				BC
PUSH IY	$(SP-2) \leftarrow IY_L$	●	●	●	●	●	●	●	11	111	101	2	4	15	01
	$(SP-1) \leftarrow IY_H$								11	100	101				DE
POP qq	$qq_H \leftarrow (SP+1)$	●	●	●	●	●	●	●	11	qq0	001	1	3	10	10
	$qq_L \leftarrow (SP)$														HL
POP IX	$IX_H \leftarrow (SP+1)$	●	●	●	●	●	●	●	11	011	101	2	4	14	11
	$IX_L \leftarrow (SP)$								11	100	001				AF
POP IY	$IY_H \leftarrow (SP+1)$	●	●	●	●	●	●	●	11	111	101	2	4	14	
	$IY_L \leftarrow (SP)$								11	100	001				

Exchange, Block Transfer and Search Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE,HL	DE ↔ HL	●	●	●	●	●	●	11	101	011	1	1	4	Each value in register pairs is exchanged with the value in aux. register pairs.
EX AF,AF'	AF ↔ AF'	●	●	●	●	●	●	00	001	000	1	1	4	
EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	●	●	●	●	●	●	11	011	001	1	1	4	
EX (SP),HL	H ↔ (SP+1) L ↔ (SP)	●	●	●	●	●	●	11	100	011	1	5	19	
EX (SP),IX	IX _H ↔ (SP+1)	●	●	●	●	●	●	11	011	101	2	6	23	
	IX _L ↔ (SP)							11	100	011				
EX (SP),IY	IY _H ↔ (SP+1)	●	●	●	●	●	●	11	111	101	2	6	23	
	IY _L ↔ (SP)							11	100	011				
LDI				①										
	(DE) ← (HL)	●	●	↑	●	0	0	11	101	101	2	4	16	
	DE ← DE+1							10	100	000				
	HL ← HL+1													
LDIR	BC ← BC-1													for BC ≠ 0 for BC = 0
	(DE) ← (HL)	●	●	0	●	0	0	11	101	101	2	5	21	
	DE ← DE+1							10	110	000	2	4	16	
	HL ← HL+1													
LDD	BC ← BC-1													
	repeats to BC=0													
				①										
	(DE) ← (HL)	●	●	↑	●	0	0	11	101	101	2	4	16	
LDDR	DE ← DE-1							10	101	000				
	HL ← HL-1													
	BC ← BC-1													
	repeats to BC=0													
CPI				②	①									
	A - (HL)	●	↑	↑	↑	1	↑	11	101	101	2	4	16	
	HL ← HL+1							10	100	001				
	BC ← BC-1													

Notes: ① indicates that the P/V flag is set to 0 for BC-1=0 and other cases to 1.

② indicates that the Z flag is set to 1 for A=(HL) and other cases to 0.

Exchange, Block Transfer and Search Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE,HL	DE ↔ HL	●	●	●	●	●	●	11	101	011	1	1	4	Each value in register pairs is exchanged with the value in aux. register pairs.
EX AF,AF'	AF ↔ AF'	●	●	●	●	●	●	00	001	000	1	1	4	
EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	●	●	●	●	●	●	11	011	001	1	1	4	
EX (SP),HL	H ↔ (SP+1) L ↔ (SP)	●	●	●	●	●	●	11	100	011	1	5	19	
EX (SP),IX	IX _H ↔ (SP+1) IX _L ↔ (SP)	●	●	●	●	●	●	11	011	101	2	6	23	
EX (SP),IY	IY _H ↔ (SP+1) IY _L ↔ (SP)	●	●	●	●	●	●	11	111	101	2	6	23	
LDI				①										
	(DE) ← (HL)	●	●	↑	●	0	0	11	101	101	2	4	16	
	DE ← DE+1							10	100	000				
	HL ← HL+1													
LDIR	BC ← BC-1													
	(DE) ← (HL)	●	●	0	●	0	0	11	101	101	2	5	21	for BC ≠ 0
	DE ← DE+1							10	110	000	2	4	16	for BC = 0
	HL ← HL+1													
LDD	BC ← BC-1													
	repeats to BC=0													
				①										
	(DE) ← (HL)	●	●	↑	●	0	0	11	101	101	2	4	16	
LDDR	DE ← DE-1							10	101	000				
	HL ← HL-1													
	BC ← BC-1													
	repeats to BC=0													
CPI				②	①									
	A - (HL)	●	↑	↑	↑	1	↑	11	101	101	2	4	16	
	HL ← HL+1							10	100	001				
	BC ← BC-1													

Notes: ① indicates that the P/V flag is set to 0 for BC-1=0 and other cases to 1.

② indicates that the Z flag is set to 1 for A=(HL) and other cases to 0.

8-Bit Arithmetic and Logical Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
ADD A,r	$A \leftarrow A + r$	↑	↑	V	↑	0	↑	↑	10	000	r	1	1	4	r
ADD A,n	$A \leftarrow A + n$	↑	↑	V	↑	0	↑	↑	11	000	110	2	2	7	000 B
									$\leftarrow n \rightarrow$						001 C
ADD A,(HL)	$A \leftarrow A + (HL)$	↑	↑	V	↑	0	↑	↑	10	000	110	1	2	7	010 D
ADD A,(IX+d)	$A \leftarrow A + (IX + d)$	↑	↑	V	↑	0	↑	↑	11	011	101	3	5	19	011 E
									10	000	110				100 H
									$\leftarrow d \rightarrow$						101 L
ADD A,(IY+d)	$A \leftarrow A + (IY + d)$	↑	↑	V	↑	0	↑	↑	11	111	101	3	5	19	111 A
									10	000	110				
									$\leftarrow d \rightarrow$						
ADC A,s	$A \leftarrow A + s + CY$	↑	↑	V	↑	0	↑	↑	001						The s operand is any of r, n, (HL), (IX+d) or (IY+d) as defined for the ADD instruction. Replace the opcode 000 in ADD with the framed bits.
SUB s	$A \leftarrow A - s$	↑	↑	V	↑	1	↑	↑	010						
SBC A,s	$A \leftarrow A - s - CY$	↑	↑	V	↑	1	↑	↑	011						
AND s	$A \leftarrow A \wedge s$	0	↑	P	↑	0	1	↑	100						
OR s	$A \leftarrow A \vee s$	0	↑	P	↑	0	0	↑	110						
XOR s	$A \leftarrow A \oplus s$	0	↑	P	↑	0	0	↑	101						
CP s	$A - s$	↑	↑	V	↑	1	↑	↑	111						
INC r	$r \leftarrow r + 1$	●	↑	V	↑	0	↑	↑	00	r	100	1	1	4	
INC (HL)	$(HL) \leftarrow (HL) + 1$	●	↑	V	↑	0	↑	↑	00	110	100	1	3	11	
INC (IX+d)	$(IX + d) \leftarrow (IX + d) + 1$	●	↑	V	↑	0	↑	↑	11	011	101	3	6	23	
									00	110	100				
									$\leftarrow d \rightarrow$						
INC (IY+d)	$(IY + d) \leftarrow (IY + d) + 1$	●	↑	V	↑	0	↑	↑	11	111	101	3	6	23	The m operand is any of r, (HL), (IX+d) or (IY+d) as defined for the INC instruction. The opcode is the same as the INC's except 100 changes to 101.
									00	110	100				
									$\leftarrow d \rightarrow$						
DEC m	$m \leftarrow m - 1$	●	↑	V	↑	1	↑	↑	101						

16-Bit Arithmetic Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P V	S	N	H	76	543	210					
ADD HL,ss	HL←HL+ss	↑	●	●	●	0	X	00	ss1	001	1	3	11	ss	register
ADC HL,ss	HL←HL+ss+CY	↑	↑	V	↑	0	X	11	101	101	2	4	15	00	BC
								01	ss1	010				01	DE
SBC HL,ss	HL←HL-ss-CY	↑	↑	V	↑	1	X	11	101	101	2	4	15	10	HL
								01	ss0	010				11	SP
ADD IX,pp	IX←IX+pp	↑	●	●	●	0	X	11	011	101	2	4	15	pp	register
								00	pp1	001				00	BC
														01	DE
														10	IX
														11	SP
ADD IY,rr	IY←IY+rr	↑	●	●	●	0	X	11	111	101	2	4	15	rr	register
								00	rr1	001				00	BC
														01	DE
														10	IY
														11	SP
INC ss	ss←ss+1	●	●	●	●	●	●	00	ss0	011	1	1	6		
INC IX	IX←IX+1	●	●	●	●	●	●	11	011	101	2	2	10		
								00	100	011					
INC IY	IY←IY+1	●	●	●	●	●	●	11	111	101	2	2	10		
								00	100	011					
DEC ss	ss←ss-1	●	●	●	●	●	●	00	ss1	011	1	1	6		
DEC IX	IX←IX-1	●	●	●	●	●	●	11	011	101	2	2	10		
								00	101	011					
DEC IY	IY←IY-1	●	●	●	●	●	●	11	111	101	2	2	10		
								00	101	011					

Rotate and Shift Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments
		C	Z	P	V	S	N	76	543	210				
RLC A		↑	●	●	●	0	0	00	000	111	1	1	4	The contents of the Accumulator are rotated left.
RL A		↑	●	●	●	0	0	00	010	111	1	1	4	
RRC A		↑	●	●	●	0	0	00	001	111	1	1	4	
RR A		↑	●	●	●	0	0	00	011	111	1	1	4	
RLC r		↑	↑	P	↑	0	0	11	001	011	2	2	8	The contents of register r are rotated left.
RLC (HL)		↑	↑	P	↑	0	0	11	001	011	2	4	15	
RLC (IX+d)		↑	↑	P	↑	0	0	11	011	101	4	6	23	
RLC (IY+d)		↑	↑	P	↑	0	0	11	111	101	4	6	23	
RL s		↑	↑	P	↑	0	0	010						
RRC s		↑	↑	P	↑	0	0	001						
RR s		↑	↑	P	↑	0	0	011						
SLA s		↑	↑	P	↑	0	0	100						
SRA s		↑	↑	P	↑	0	0	101						
SRL s		↑	↑	P	↑	0	0	111						
RLD		●	↑	P	↑	0	0	11	101	101	2	5	18	The s operand is any of r, (HL), (IX+d) or (IY+d)
RRD		●	↑	P	↑	0	0	11	101	101	2	5	18	

Bit Set, Reset and Test Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P/V	S	N	H	76	543	210					
BIT b,r	$Z \leftarrow \bar{r}_b$	•	↓	X	X	0	1	11	001	011	2	2	8	r	register
BIT b,(HL)	$Z \leftarrow \overline{(HL)}_b$	•	↓	X	X	0	1	01	b	r	2	3	12	000	B
								11	001	011				001	C
BIT b,(IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	•	↓	X	X	0	1	01	b	110	4	5	20	010	D
								11	011	101				011	E
								11	001	011				100	H
								←	d	→				101	L
BIT b,(IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	•	↓	X	X	0	1	01	b	110	4	5	20	111	A
								11	111	101				b	bit tested
								11	001	011				000	0
								←	d	→				001	1
SET b,r	$\bar{r}_b \leftarrow 1$	•	•	•	•	•	•	01	b	110	2	2	8	010	2
								11	001	011				011	3
SET b,(HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11	b	r	2	4	15	100	4
								11	001	011				101	5
SET b,(IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	11	b	110	4	6	23	110	6
								11	011	101				111	7
								11	001	011					
								←	d	→					
SET b,(IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	11	b	110	4	6	23		
								11	111	101					
								11	001	011					
								←	d	→					
RES b,s	$s_b \leftarrow 0$ $s \equiv r, (HL),$ $(IX+d),$ $(IY+d)$	•	•	•	•	•	•	11	b	110				Bit b in operand s is reset	
								10							

Jump Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P/V	S	N	H	76	543	210					
JP nn	PC←nn	●	●	●	●	●	●	11	000	011	3	3	10		
								←	n	→					
								←	n	→					
JP cc,nn	if cc is true, PC←nn if false, continue	●	●	●	●	●	●	11	cc	010	3	3	10	cc	condition
								←	n	→				000	NZ non zero
								←	n	→				001	Z zero
JR e	PC←PC+e	●	●	●	●	●	●	00	011	000	2	3	12	010	NC non carry
								←	e-2	→				011	C carry
JR C,e	if C=0, continue	●	●	●	●	●	●	00	111	000	2	2	7	100	PO parity odd
								←	e-2	→				101	PE parity even
	if C=1, PC←PC+e										2	3	12	110	P sign positive
JR NC,e	if C=1, continue	●	●	●	●	●	●	00	110	000	2	2	7	111	M sign negative
	if C=0 PC←PC+e										2	3	12		
JR Z,e	if Z=0, continue	●	●	●	●	●	●	00	101	000	2	2	7		
								←	e-2	→					
	if Z=1, PC←PC+e										2	3	12		
JR NZ,e	if Z=1, continue	●	●	●	●	●	●	00	100	000	2	2	7		
								←	e-2	→					
	if Z=0 PC←PC+e										2	3	12		
JP (HL)	PC←HL	●	●	●	●	●	●	11	101	001	1	1	4		
JP (IX)	PC←IX	●	●	●	●	●	●	11	011	101	2	2	8		
								11	101	001					
JP (IY)	PC←IY	●	●	●	●	●	●	11	111	101	2	2	8		
								11	101	001					
DJNZ,e	B←B-1 if B=0, continue	●	●	●	●	●	●	00	010	000	2	2	8	for B=0	
								←	e-2	→					
	if B≠0 PC←PC+e										2	3	13	for B≠0	

Note: The value of the displacement e has a range of -126 to +129 bytes. the binnary number equivalent to e-2 must be placed in opcode.

Call and Return Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P/V	S	N	H	76	543	210					
CALL nn	(SP-1)←PC _H	•	•	•	•	•	•	11	001	101	3	5	17		
	(SP-2)←PC _L							←	n	→					
	PC←nn							←	n	→					
CALL cc,nn	if cc is true,	•	•	•	•	•	•	11	cc	100	3	3	10	for cc is false	
	equal to CALL nn							←	n	→					
	if false, continue							←	n	→	3	5	17	for cc is true	
RET	PC _L ←(SP)	•	•	•	•	•	•	11	001	001	1	3	10		
	PC _H ←(SP+1)														
RET cc	if cc is true,	•	•	•	•	•	•	11	cc	000	1	1	5	for cc is false	
	equal to RET										1	3	11	for cc is true	
	if false, continue													cc	condition
RETI	return from interrupt	•	•	•	•	•	•	11	101	101	2	4	14	000	NZ non zero
								01	001	101				001	Z zero
RETN	return from NMI	•	•	•	•	•	•	11	101	101	2	4	14	010	NC non carry
								01	000	101				011	C carry
RST p	(SP-1)←PC _H (SP-2)←PC _L PC _H ←0 PC _L ←P	•	•	•	•	•	•	11	t	111	1	3	11	100	PO parity odd
														101	PE parity even
														110	P sign positive
														111	M sign negative
														t	P
														000	00H
														001	08H
														010	10H
														011	18H
														100	20H
														101	28H
														110	30H
														111	38H

object code	mnemonic
4C	LD C, H
4D	LD C, L
4E	LD C, (HL)
4F	LD C, A
50	LD D, B
51	LD D, C
52	LD D, D
53	LD D, E
54	LD D, H
55	LD D, L
56	LD D, (HL)
57	LD D, A
58	LD E, B
59	LD E, C
5A	LD E, D
5B	LD E, E
5C	LD E, H
5D	LD E, L
5E	LD E, (HL)
5F	LD E, A
60	LD H, B
61	LD H, C
62	LD H, D
63	LD H, E
64	LD H, H
65	LD H, L
66	LD H, (HL)
67	LD H, A
68	LD L, B
69	LD L, C
6A	LD L, D
6B	LD L, E
6C	LD L, H
6D	LD L, L
6E	LD L, (HL)
6F	LD L, A
	LD (HL), B

object code	mnemonic
71	LD (HL), C
72	LD (HL), D
73	LD (HL), E
74	LD (HL), H
75	LD (HL), L
76	HALT
77	LD (HL), A
78	LD A, B
79	LD A, C
7A	LD A, D
7B	LD A, E
7C	LD A, H
7D	LD A, L
7E	LD A, (HL)
7F	LD A, A
80	ADD A, B
81	ADD A, C
82	ADD A, D
83	ADD A, E
84	ADD A, H
85	ADD A, L
86	ADD A, (HL)
87	ADD A, A
88	ADC A, B
89	ADC A, C
8A	ADC A, D
8B	ADC A, E
8C	ADC A, H
8D	ADC A, L
8E	ADC A, (HL)
8F	ADC A, A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB (HL)

object code	mnemonic
97	SUB A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
9E	SBC A,(HL)
9F	SBC A,A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
A6	AND (HL)
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H

object code	mnemonic
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
<u>C28405</u>	JP NZ,nn
<u>C38405</u>	JP nn
<u>C48405</u>	CALL NZ,nn
C5	PUSH BC
<u>C620</u>	ADD A,n
C7	RST 0
C8	RET Z
C9	RET
<u>CA8405</u>	JP Z,nn
<u>CC8405</u>	CALL Z,nn
<u>CD8405</u>	CALL nn
<u>CE20</u>	ADC A,n
CF	RST 1
D0	RET NC
D1	POP DE
<u>D28405</u>	JP NC,nn
<u>D320</u>	OUT (n),A
<u>D48405</u>	CALL NC,nn
D5	PUSH DE
<u>D620</u>	SUB n
D7	RST 2
D8	RET C
D9	EXX
<u>DA8405</u>	JP C,nn
<u>DB20</u>	IN A,(n)
<u>DC8405</u>	CALL C,nn
<u>DE20</u>	SBC A,n
DF	RST 3
E0	RET PO
E1	POP HL
<u>E28405</u>	JP PO,nn
E3	EX (SP),HL

object code	mnemonic
E4 <u>8405</u>	CALL PO,nn
E5	PUSH HL
E6 <u>20</u>	AND n
E7	RST 4
E8	RET PE
E9	JP (HL)
EA <u>8405</u>	JP PE,nn
EB	EX DE,HL
EC <u>8405</u>	CALL PE,nn
EE <u>20</u>	XOR n
EF	RST 5
F0	RET P
F1	POP AF
F2 <u>8405</u>	JP P,nn
F3	DI
F4 <u>8405</u>	CALL P,nn
F5	PUSH AF
F6 <u>20</u>	OR n
F7	RST 6
F8	RET M
F9	LD SP,HL
FA <u>8405</u>	JP M,nn
FB	EI
FC <u>8405</u>	CALL M,nn
FE <u>20</u>	CP n
FF	RST 7
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
CB06	RLC (HL)
CB07	RLC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E

object code	mnemonic
CB0C	RRC H
CB0D	RRC L
CB0E	RRC (HL)
CB0F	RRC A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
CB16	RL (HL)
CB17	RL A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
CB1E	RR (HL)
CB1F	RR A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB26	SLA (HL)
CB27	SLA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB2E	SRA (HL)
CB2F	SRA A
CB38	SRL B

object code	mnemonic
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL (HL)
CB3F	SRL A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB46	BIT 0,(HL)
CB47	BIT 0,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB4E	BIT 1,(HL)
CB4F	BIT 1,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB56	BIT 2,(HL)
CB57	BIT 2,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB5E	BIT 3,(HL)

object code	mnemonic
CB5F	BIT 3,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB66	BIT 4,(HL)
CB67	BIT 4,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB6E	BIT 5,(HL)
CB6F	BIT 5,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB76	BIT 6,(HL)
CB77	BIT 6,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
CB7E	BIT 7,(HL)
CB7F	BIT 7,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E

object code	mnemonic
CB84	RES 0, H
CB85	RES 0, L
CB86	RES 0, (HL)
CB87	RES 0, A
CB88	RES 1, B
CB89	RES 1, C
CB8A	RES 1, D
CB8B	RES 1, E
CB8C	RES 1, H
CB8D	RES 1, L
CB8E	RES 1, (HL)
CB8F	RES 1, A
CB90	RES 2, B
CB91	RES 2, C
CB92	RES 2, D
CB93	RES 2, E
CB94	RES 2, H
CB95	RES 2, L
CB96	RES 2, (HL)
CB97	RES 2, A
CB98	RES 3, B
CB99	RES 3, C
CB9A	RES 3, D
CB9B	RES 3, E
CB9C	RES 3, H
CB9D	RES 3, L
CB9E	RES 3, (HL)
CB9F	RES 3, A
CBA0	RES 4, B
CBA1	RES 4, C
CBA2	RES 4, D
CBA3	RES 4, E
CBA4	RES 4, H
CBA5	RES 4, L
CBA6	RES 4, (HL)
CBA7	RES 4, A
CBA8	RES 5, B
CBA9	RES 5, C

object code	mnemonic
CBA A	RES 5, D
CBAB	RES 5, E
CBAC	RES 5, H
CBAD	RES 5, L
CBAE	RES 5, (HL)
CBAF	RES 5, A
CBB0	RES 6, B
CBB1	RES 6, C
CBB2	RES 6, D
CBB3	RES 6, E
CBB4	RES 6, H
CBB5	RES 6, L
CBB6	RES 6, (HL)
CBB7	RES 6, A
CBB8	RES 7, B
CBB9	RES 7, C
CBBA	RES 7, D
CBBB	RES 7, E
CBBC	RES 7, H
CBB D	RES 7, L
CBBE	RES 7, (HL)
CBBF	RES 7, A
CBC0	SET 0, B
CBC1	SET 0, C
CBC2	SET 0, D
CBC3	SET 0, E
CBC4	SET 0, H
CBC5	SET 0, L
CBC6	SET 0, (HL)
CBC7	SET 0, A
CBC8	SET 1, B
CBC9	SET 1, C
CBCA	SET 1, D
CBCB	SET 1, E
CBCC	SET 1, H
CBCD	SET 1, L
CBCE	SET 1, (HL)
CBCF	SET 1, A

object code	mnemonic
CBD0	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD6	SET 2,(HL)
CBD7	SET 2,A
CBD8	SET 3,B
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBDE	SET 3,(HL)
CBDF	SET 3,A
CBE0	SET 4,B
CBE1	SET 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBE6	SET 4,(HL)
CBE7	SET 4,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBEE	SET 5,(HL)
CBEF	SET 5,A
CBF0	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L

object code	mnemonic
CBF6	SET 6,(HL)
CBF7	SET 6,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CBFE	SET 7,(HL)
CBFF	SET 7,A
DD09	ADD IX,BC
DD19	ADD IX,DE
DD21 ⁸⁴⁰⁵	LD IX,nn
DD22 ⁸⁴⁰⁵	LD (nn),IX
DD23	INC IX
DD29	ADD IX,IX
DD2A ⁸⁴⁰⁵	LD IX,(nn)
DD2B	DEC IX
DD34 ⁰⁵	INC (IX+d)
DD35 ⁰⁵	DEC (IX+d)
DD3605 ⁰⁵²⁰	LD (IX+d),n
DD39	ADD IX,SP
DD46 ⁰⁵	LD B,(IX+d)
DD4E ⁰⁵	LD C,(IX+d)
DD56 ⁰⁵	LD D,(IX+d)
DD5E ⁰⁵	LD E,(IX+d)
DD66 ⁰⁵	LD H,(IX+d)
DD6E ⁰⁵	LD L,(IX+d)
DD70 ⁰⁵	LD (IX+d),B
DD71 ⁰⁵	LD (IX+d),C
DD72 ⁰⁵	LD (IX+d),D
DD73 ⁰⁵	LD (IX+d),E
DD74 ⁰⁵	LD (IX+d),H
DD75 ⁰⁵	LD (IX+d),L
DD77 ⁰⁵	LD (IX+d),A
DD7E ⁰⁵	LD A,(IX+d)
DD86 ⁰⁵	ADD A,(IX+d)
DD8E ⁰⁵	ADC A,(IX+d)
DD96 ⁰⁵	SUB (IX+d)

object code	mnemonic
DD9E05	SBC A,(IX+d)
DDA605	AND (IX+d)
DDAE05	XOR (IX+d)
DDb605	OR (IX+d)
DDBE05	CP (IX+d)
DDE1	POP IX
DDE3	EX (SP),IX
DDE5	PUSH IX
DDE9	JP (IX)
DDF9	LD SP,IX
DDCB0506	RLC (IX+d)
DDCB050E	RRC (IX+d)
DDCB0516	RL (IX+d)
DDCB051E	RR (IX+d)
DDCB0526	SLA (IX+d)
DDCB052E	SRA (IX+d)
DDCB053E	SRL (IX+d)
DDCB0546	BIT 0,(IX+d)
DDCB054E	BIT 1,(IX+d)
DDCB0556	BIT 2,(IX+d)
DDCB055E	BIT 3,(IX+d)
DDCB0566	BIT 4,(IX+d)
DDCB056E	BIT 5,(IX+d)
DDCB0576	BIT 6,(IX+d)
DDCB057E	BIT 7,(IX+d)
DDCB0586	RES 0,(IX+d)
DDCB058E	RES 1,(IX+d)
DDCB0596	RES 2,(IX+d)
DDCB059E	RES 3,(IX+d)
DDCB05A6	RES 4,(IX+d)
DDCB05AE	RES 5,(IX+d)
DDCB05B6	RES 6,(IX+d)
DDCB05BE	RES 7,(IX+d)
DDCB05C6	SET 0,(IX+d)
DDCB05CE	SET 1,(IX+d)
DDCB05D6	SET 2,(IX+d)
DDCB05DE	SET 3,(IX+d)
DDCB05E6	SET 4,(IX+d)
DDCB05EE	SET 5,(IX+d)

object code	mnemonic
DDCB05F6	SET 6,(IX+d)
DDCB05FE	SET 7,(IX+d)
ED40	IN B,(C)
ED41	OUT (C),B
ED42	SBC HL,BC
ED438405	LD (nn),BC
ED44	NEG
ED45	RETN
ED46	IM 0
ED47	LD I,A
ED48	IN C,(C)
ED49	OUT (C),C
ED4A	ADC HL,BC
ED4B8405	LD BC,(nn)
ED4D	RETI
ED50	IN D,(C)
ED51	OUT (C),D
ED52	SBC HL,DE
ED538405	LD (nn),DE
ED56	IM 1
ED57	LD A,I
ED58	IN E,(C)
ED59	OUT (C),E
ED5A	ADC HL,DE
ED5B8405	LD DE,(nn)
ED5E ED5F	IM 2 LD A,R
ED60	IN H,(C)
ED61	OUT (C),H
ED62	SBC HL,HL
ED67	RRD
ED68	IN L,(C)
ED69	OUT (C),L
ED6A	ADC HL,HL
ED6F	RLD
ED72	SBC HL,SP
ED738405	LD (nn),SP
ED78	IN A,(C)
ED79	OUT (C),A
ED7A	ADC HL,SP

object code	mnemonic
ED7B ⁸⁴⁰⁵	LD SP,(nn)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR
FD09	ADD IY,BC
FD19	ADD IY,DE
FD21 ⁸⁴⁰⁵	LD IY,nn
FD22 ⁸⁴⁰⁵	LD (nn),IY
FD23	INC IY
FD29	ADD IY,IY
FD2A ⁸⁴⁰⁵	LD IY,(nn)
FD2B	DEC IY
FD34 ⁰⁵	INC (IY+d)
FD35 ⁰⁵	DEC (IY+d)
FD36 ⁰⁵²⁰	LD (IY+d),n
FD39	ADD IY,SP
FD46 ⁰⁵	LD B,(IY+d)
FD4E ⁰⁵	LD C,(IY+d)
FD56 ⁰⁵	LD D,(IY+d)
FD5E ⁰⁵	LD E,(IY+d)
FD66 ⁰⁵	LD H,(IY+d)
FD6E ⁰⁵	LD L,(IY+d)
FD70 ⁰⁵	LD (IY+d),B
FD71 ⁰⁵	LD (IY+d),C
FD72 ⁰⁵	LD (IY+d),D
FD73 ⁰⁵	LD (IY+d),E

object code	mnemonic
FD74 ⁰⁵	LD (IY+d),H
FD75 ⁰⁵	LD (IY+d),L
FD77 ⁰⁵	LD (IY+d),A
FD7E ⁰⁵	LD A,(IY+d)
FD86 ⁰⁵	ADD A,(IY+d)
FD8E ⁰⁵	ADC A,(IY+d)
FD96 ⁰⁵	SUB (IY+d)
FD9E ⁰⁵	SBC A,(IY+d)
FDA6 ⁰⁵	AND (IY+d)
FDAE ⁰⁵	XOR (IY+d)
FDB6 ⁰⁵	OR (IY+d)
FDBE ⁰⁵	CP (IY+d)
FDE1	POP IY
FDE3	EX (SP),IY
FDE5	PUSH IY
FDE9	JP,(IY)
FDF9	LD SP,IY
FDCB ⁰⁵⁰⁶	RLC (IY+d)
FDCB ^{050E}	RRC (IY+d)
FDCB ⁰⁵¹⁶	RL (IY+d)
FDCB ^{051E}	RR (IY+d)
FDCB ⁰⁵²⁶	SLA (IY+d)
FDCB ^{052E}	SRA (IY+d)
FDCB ^{053E}	SRL (IY+d)
FDCB ⁰⁵⁴⁶	BIT 0,(IY+d)
FDCB ^{054E}	BIT 1,(IY+d)
FDCB ⁰⁵⁵⁶	BIT 2,(IY+d)
FDCB ^{055E}	BIT 3,(IY+d)
FDCB ⁰⁵⁶⁶	BIT 4,(IY+d)
FDCB ^{056E}	BIT 5,(IY+d)
FDCB ⁰⁵⁷⁶	BIT 6,(IY+d)
FDCB ^{057E}	BIT 7,(IY+d)
FDCB ⁰⁵⁸⁶	RES 0,(IY+d)
FDCB ^{058E}	RES 1,(IY+d)
FDCB ⁰⁵⁹⁶	RES 2,(IY+d)
FDCB ^{059E}	RES 3,(IY+d)
FDCB ^{05A6}	RES 4,(IY+d)
FDCB ^{05AE}	RES 5,(IY+d)
FDCB ^{05B6}	RES 6,(IY+d)

object code	mnemonic
FDCB <u>05</u> BE	RES 7,(IY+d)
FDCB <u>05</u> C6	SET 0,(IY+d)
FDCB <u>05</u> CE	SET 1,(IY+d)
FDCB <u>05</u> D6	SET 2,(IY+d)
FDCB <u>05</u> DE	SET 3,(IY+d)
FDCB <u>05</u> E6	SET 4,(IY+d)
FDCB <u>05</u> EE	SET 5,(IY+d)
FDCB <u>05</u> F6	SET 6,(IY+d)
FDCB <u>05</u> FE	SET 7,(IY+d)

Value Examples :

nn, n, d and e are the value examples.

where :

nn=584H

n= 20H

d= 5

e= 30H

In the object code column, the code equivalent to this value is indicated in *italics* and by an underline.

INDEX

[A]

access, read/write	21
access timing, VRAM	13
add/subtract flag	138
ALPHA ROTATE	41
ALL INITIALIZE	41
ASC	54
AXIS (AXIS)	41

[B]

back space	40
BELL	52
bit set	149
block transfer	144
border color	23
BRKEY	53

[C]

call (instruction)	151
carriage return	40
carry flag	138
cassette control	26
cassette recorder circuit	27
C.G. ROM, modify	45
C.G. table	46
character generator	45
character scale	40
character set	37
character set, table	39
clock generator	9
COLOR CHANGE (PCOLOR)	41
color position detector	37
CPU control (instruction)	145
custom LSI	6
custom LSI block diagram	8
cycle steal mode	15

[D]

DA (display address)	12
disk control	100
display address generator	9
display mode register (DMD)	17
display period	16
display timing	14
DMA (display memory address)	12
DRAW (LINE)	41

[E]

ELMD	55
ELMD1	55
ELMD20	55
ELMD22	55
ELMD24	55
ERRORP	55
exchange	144

[F]

FILOUT	55
flags	138
flyback period	16
FONT	45
form feed	40

[G]

general purpose arithmetic (instruction)	145
GETKY	53
GETL	53
graphic code	40
graphic mode	41

[H]

half carry flag	139
HLHEX	54
HOME (PHONE)	41

[I]

INITIALIZE (HSET)	41
input (instruction)	152

[J]

joystick	32
jump (instruction)	150

[K]

key scan	25
----------------	----

[L]

LETNL	52
line counter set	40
line feed	40
line scroll	10

LINE TYPE	41
line up	40
logical (instruction)	146

[M]

MA	12
magnify scale	40
MELDY	52
memory bank	4
memory controller	6
memory map	3
monitor <1Z-013B>	67
monitor call	55
monitor subroutines	52
monitor variable	55
MOVE (MOVE)	41
MSG	52
MSTA	52
MSTP	53
MZ1P16	35
MZ1P16, specification	42

[N]

next colour	40
notation	140

[O]

output (instruction)	152
overflow (flag)	139

[P]

pallet	22
parity/overflow flag	139
pen exchange method	37
pen exchange operation	36
pen test	40
plotter printer control codes	40
plotter printer control LSI	38
POOL	55
power supply	33
PRINT	41
PRNTS	52
pseud cycle steal	13

[R]

range of scroll	12
-----------------------	----

read format register	19
read/write access	21
reduction cancel	40
reduction scale	40
RELATIVE DRAW (RLINE)	41
RELATIVE MOVE (RMOVE)	41
reset (instruction)	149
return (instruction)	151
ROM configuration	24
ROM monitor	52
ROM, C.G.	45
rotate (instruction)	148

[S]

SCALE SET	41
screen split	10
scroll	10,11
scroll method	12
scroll sequence	12
scroll, range of	12
scrolling, address conversion	12
SEA	10,11
search (instruction)	144
shift (instruction)	148
sign flag	139
special key (read with GETKEY)	53
specification, MZ1P16	42
SSA	10,11
status indicators	138
stepping motor	37
subtract (flag)	138
SW	10,11
SYSSTA	55
system description	1
system switches	32

[T]

TEMLMT	55
test (instruction)	149
text code	40
TEXTST	55
timing generator	9
TIMRD	53
TIMST	53
TMPEND	55

[V]

VARST	55
VCAS	16
VRAM access timing	13
VRAM accessing	16
VRAM configuration	17
VRAM, I/O circuit	12,13

[W]

write format register (WR)	20
----------------------------------	----

[X]

XTEMP	52
-------------	----

[Z]

Z-80 programming reference	138
zero flag	138
ZLOG	55
ZRWX	55

[Others]

16-bit arithmetic (instruction)	147
16-bit load (instruction)	142
2HEX	54
8-bit arithmetic (instruction)	146
8-bit load (instruction)	141
8050, pin configuration	37
8253	28
8255	24,25
.BREAK	58
.CLKL	61
.COUNT	58
.CRTMS	56
.CRT1C	56
.CRT1X	56
.CR1	55
.CR2	56
.DEVFN	59
.DEVNM	59
.DI	58
.DIR	61
.EI	58
.ERCVR	62
.ERRX	59
.FINFT	62
.GETL	57
.HALT	58
.INKEY	58

.INMSG	61
.LOADF	60
.LOPEN	59
.LPTOT	56
.LPT1C	56
.LUCHK	59
.MONOP	55
.PRSTR	61
.RWOPN	60
.SAVEF	60
.SETDF	62
.VRFYF	60
.&CR	57
.&MSG	57
.&1C	57
.&1CX	57
?ADCN	54
?BLNK	54
?DACN	54
?DPCT	54
?POINT	54
??KEY	54

SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
Gedruckt in Japan
Imprimé au Japon
Stampato in Giappone
©1985 SHARP CORPORATION
5E 2.25-S(TINSE1318ACZZ)①